

Rec'd PCT/PTO 21 JAN 2005

JP03/13405

72

PCT/JP03/13405

10/521789

日本国特許庁
JAPAN PATENT OFFICE

21.10.03 RECEIVED
04 DEC 2003
WIPO PCT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2003年 9月17日

出願番号
Application Number: 特願2003-324805
[ST. 10/C]: [JP2003-324805]

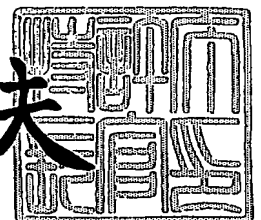
出願人
Applicant(s): 松下電器産業株式会社

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2003年11月21日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



出証番号 出証特2003-3096448

【書類名】 特許願
【整理番号】 2931050083
【提出日】 平成15年 9月17日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 9/06
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 里 雄二
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 山口 孝雄
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 佐藤 潤一
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 武井 一朗
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 伊藤 智祥
【特許出願人】
 【識別番号】 000005821
 【氏名又は名称】 松下電器産業株式会社
【代理人】
 【識別番号】 100105050
 【弁理士】
 【氏名又は名称】 鷲田 公一
【先の出願に基づく優先権主張】
 【出願番号】 特願2002-311815
 【出願日】 平成14年10月25日
【先の出願に基づく優先権主張】
 【出願番号】 特願2003-133566
 【出願日】 平成15年 5月12日
【手数料の表示】
 【予納台帳番号】 041243
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9700376

【書類名】 特許請求の範囲**【請求項 1】**

プログラムの配布先ごとに異なる透かし情報を前記プログラムに挿入する透かし情報挿入手段と、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させない透かし検証コードを前記プログラムに挿入するコード挿入手段と、を具備し、

前記透かし検証コードを前記配布先に関わらず同じにしたことを特徴とする透かし挿入装置。

【請求項 2】

前記透かし情報を、プログラムの配布先を一意に決定する ID 情報から生成することを特徴とする請求項 1 記載の透かし挿入装置。

【請求項 3】

前記透かし情報から所定の定数を入力する関数を定義し、前記関数を変数に代入する式を前記プログラムに挿入する関数挿入手段を具備し、

前記透かし検証コードは、前記変数と前記定数が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐であり、

前記定数を前記配布先に関わらず同じにしていることを特徴とする請求項 1 または請求項 2 記載の透かし挿入装置。

【請求項 4】

前記透かし検証コードは、前記プログラムを正しく動作させるのに必要なものであることを特徴とする請求項 1 または請求項 2 記載の透かし挿入装置。

【請求項 5】

前記透かし検証コードは、前記プログラムから取り出した判定分岐に前記透かし情報から生成した前記判定分岐の判定文に影響を与えない計算式を挿入したものであることを特徴とする請求項 4 記載の透かし挿入装置。

【請求項 6】

請求項 1 から請求項 5 のいずれかに記載の透かし挿入装置が前記透かし情報および前記透かし検証コードを挿入したプログラムを入力するプログラム入力手段と、前記プログラムから前記透かし情報を取り出し、前記透かし情報に基づいて前記配布先を一意に特定する ID 情報を生成する透かし検出手段と、を具備し、

生成した前記 ID 情報に基づき配布先を特定することを特徴とする透かし取出装置。

【請求項 7】

請求項 1 から請求項 5 のいずれかに記載の透かし挿入装置と、請求項 6 に記載の透かし取出装置と、を具備したことを特徴とするプログラム不正配布防止システム。

【請求項 8】

前記透かし挿入装置を前記配布先に設けたことを特徴とする請求項 7 記載のプログラム不正配布防止システム。

【請求項 9】

配布先ごとに異なる透かし情報を前記プログラムに挿入するステップと、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させないものであり、前記配布先に関わらず同じ透かし検証コードを前記プログラムに挿入するステップと、を具備したことを特徴とする透かし挿入方法。

【請求項 10】

プログラムの配布先ごとに異なる透かし情報をプログラムに挿入するステップと、前記透かし情報の挿入箇所の周辺もしくは前記プログラムの全体を、仕様を維持したまま変換するステップと、を有することを特徴とする透かし挿入方法。

【請求項 11】

コンピュータに、プログラムの配布先ごとに異なる透かし情報を前記プログラムに挿入するステップと、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記配布用プログラムを正しく動作させないものであり、前記配布先に関わらず同じ透かし検証コードを前記配布用プログラムに挿入するステップと、を行わせること

を特徴とする透かし挿入プログラム。

【請求項 12】

プログラムの配布先ごとに異なる透かし情報をプログラムに挿入する透かし挿入手段と、前記透かし情報を挿入する箇所以外の部分を前記プログラムの仕様を、維持したまま変換する変換手段と、を具備したことを特徴とする透かし挿入装置。

【請求項 13】

前記変換手段は、前記透かし情報を挿入する箇所以外の部分に、仕様に影響を与えない実行コードの組を挿入することを特徴とする請求項 12 記載の透かし挿入装置。

【請求項 14】

前記透かし情報の挿入箇所を示す識別情報を記憶することを特徴とする請求項 12 または請求項 13 に記載の透かし挿入装置。

【請求項 15】

前記識別情報は、メソッド名もしくは行番号であることを特徴とする請求項 14 に記載の透かし挿入装置。

【請求項 16】

前記変換手段は、前記透かし情報を挿入する箇所以外の部分に、仕様に影響を与えないように難読化処理することを特徴とする請求項 12 記載の透かし挿入装置。

【請求項 17】

請求項 12 から請求項 16 のいずれかに記載の透かし挿入装置が前記透かし情報を挿入したプログラムを入力するプログラム入力手段と、前記プログラムから前記透かし情報を取り出す透かし検出手段と、を具備し、

取り出した前記透かし情報に基づき配布先を特定することを特徴とする透かし取出装置。

【請求項 18】

請求項 15 または請求項 16 に記載の透かし挿入装置が前記透かし情報を挿入したプログラムを入力するプログラム入力手段と、前記識別情報を取得し、前記識別情報から透かし挿入箇所を特定し、特定した前記透かし挿入箇所のみから前記透かし情報を取り出す透かし検出手段と、を具備し、

取り出した前記透かし情報に基づき配布先を特定することを特徴とする透かし取出装置。

【請求項 19】

コンピュータに、プログラムの配布先ごとに異なる透かし情報をプログラムに挿入するステップと、前記透かし情報を挿入する箇所以外の部分を前記プログラムの仕様を変更することなく変換するステップと、行わせることを特徴とするプログラム。

【請求項 20】

前記変換手段は、前記透かし情報を挿入する箇所以外の部分であって、順序を入れ替えても仕様に影響を与えない部分の順序を変換することを特徴とする請求項 12 記載の透かし挿入装置。

【請求項 21】

前記仕様に影響を与えない部分を変換した履歴情報を保持し、前記履歴情報を用いて、前記仕様に影響を与えない部分の変換を配布先ごとに異なるようにすることを特徴とする請求項 20 記載の透かし挿入装置。

【書類名】 明細書**【発明の名称】 透かし挿入装置および透かし取出装置****【技術分野】****【0001】**

本発明は、プログラムの不正な使用および配布を防止および抑止するためのプログラムへの透かしの挿入装置および透かし取出装置に関するものである。

【背景技術】**【0002】**

コンピュータネットワークの進展に伴い、ネットワークを介したコンピュータプログラムの流通が一般的になっている。コンピュータプログラムは、容易に複製を作成できるため、プログラムの複製が不正に2次配布されたり、プログラム中のアルゴリズムを盗用、改ざんされたりする可能性がある。したがって、このような不正利用からプログラムを保護する必要がある。

【0003】

従来のプログラム保護の技術の一つとして、プログラムへ電子透かしを挿入する方法が挙げられる。この方法では、配布先ごとに異なる透かし情報をプログラムに埋め込んで配布する。そして、不正利用が発生した場合に、不正利用者のプログラムから透かし情報を取り出し、透かし情報を解析する。これにより、流出元を容易に検出することが可能となる。

【0004】

具体的な透かしの挿入方法としては、たとえば、特許文献1に開示されたものがある。

【0005】

この方法は、まず、実行順序に依存関係のないコードを検出する。次に、検出部分にダミー変数の演算を挿入する。そして、ダミー変数の演算を含む検出部分の実行順序をランダムに入れ替える。

【0006】

このような処理を行うことにより、その実行順序を電子透かし情報として配布先ごとに変更する仕組みを実現している。

【特許文献1】 特開 2000-76064 号公報（第3-4頁、第2図、第7図）

【発明の開示】**【発明が解決しようとする課題】****【0007】**

しかしながら、従来のプログラムへの電子透かし挿入方式は、差分攻撃に基づく透かしの改変、削除が容易であるという問題がある。

【0008】

差分攻撃とは、複数の電子透かしの挿入されたプログラムの差分をとることで、透かしデータの挿入箇所を特定する攻撃法である。

【0009】

従来の方式を用いて、プログラムに配布先ごとに異なる透かし情報を挿入した場合、各配布先に配布されたプログラムの間で差分をとると、透かしの挿入された個所だけが差分として浮かび上がってしまう。このように、差分攻撃で透かしの挿入位置が簡単に特定されてしまい、透かし情報の削除、改ざんが容易に可能であるという問題がある。

【0010】

本発明は、かかる点に鑑みてなされたものであり、透かしの挿入箇所を特定されないように透かしを挿入することにより、透かしがなく、かつ正常に動作するプログラムを容易に生成できないようにすることを目的とする。

【課題を解決するための手段】**【0011】**

上記の課題を解決するための本発明は、プログラムの配布先を一意に特定するID情報から透かし情報を生成し、生成した透かし情報をプログラムに挿入し、透かし情報が改ざ

んされた場合にはプログラムを正しく動作させないものであり、かつ配布先に関わらず同じ透かし検証コードをプログラムに挿入するようにした。

【発明の効果】

【0012】

これにより、差分攻撃により透かしである透かし検証コードが検出されないようにできる。この結果、配布先は、透かしがなく、かつ正常に動作するプログラムを生成できないので、プログラムを不正に流通させることができなくなる。

【0013】

また、透かし情報を挿入したあとに、透かし挿入箇所周辺や、プログラム全体をプログラムの仕様を変更しない範囲で配布先ごとに異なるように変換するようにした。

【0014】

これにより、プログラムの差分を取った際に、透かし情報以外の部分が差分として浮かび上がるため、透かし挿入箇所を容易に特定することができなくなる。

【発明を実施するための最良の形態】

【0015】

本発明の第1の態様にかかる透かし挿入装置は、プログラムの配布先ごとに異なる透かし情報を前記プログラムに挿入する透かし情報挿入手段と、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させない透かし検証コードを前記プログラムに挿入するコード挿入手段と、を具備し、前記透かし検証コードを前記配布先に関わらず同じにしたものである。

【0016】

このように透かし検証コードを配布先に関わらず同じにすることにより、差分攻撃により、透かし検証コードが差分として検出されないようにできる。これにより、差分攻撃により検出された箇所だけを改変、削除するという単純な手法では、透かし検証コードを改変、削除できなくなる。よって、配布先は、透かしがなく、かつ正常に動作するプログラムを生成できないので、プログラムを不正に流通させることができなくなる。

【0017】

本発明の第2の態様は、第1の態様にかかる透かし挿入装置において、配布先を一意に決定するID情報に基づき透かし情報を生成する。

【0018】

これにより、配布先が不正配布を行った際に、配布先を一意に特定することができ、プログラムの不正流通を防止できる。

【0019】

本発明の第3の態様は、第1の態様または第2の態様にかかる透かし挿入装置において、前記透かし情報から所定の定数を出力する関数を定義し、前記関数を変数に代入する式を前記プログラムに挿入する関数挿入手段を具備し、前記透かし検証コードは、前記変数と前記定数が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐であり、前記定数を前記配布先に関わらず同じにしている。

【0020】

このような条件分布は、差分攻撃により検出されないので、差分攻撃により検出された箇所だけを改変、削除するという単純な手法では、全ての透かしを改変、削除できなくなる。よって、配布先は、透かしがなく、かつ正常に動作するプログラムを生成できないので、プログラムを不正に流通させることができなくなる。

【0021】

本発明の第4の態様は、第1の態様または第2の態様にかかる透かし挿入装置において、前記透かし検証コードは、前記プログラムを正しく動作させるのに必要なものである。

【0022】

これにより、差分攻撃により検出した透かし情報から透かし検証コードを検出し、透かし検証コードを削除、改変してしまうとプログラムが正常に動作しないようになる。つまり、正常に動作する透かし情報のない（もしくは改変された）プログラムを生成すること

を不可能にすることができるので、プログラムの不正配布を防止できる。

【0023】

本発明の第5の態様は、第4の態様にかかる透かし挿入装置において、前記透かし検証コードは、前記プログラムから取り出した判定分岐に前記透かし情報から生成した前記判定分岐の判定文に影響を与えない計算式を挿入したものである。

【0024】

これにより、プログラムを正しく動作させるのに必要な透かし検証コードを入力できる。

【0025】

本発明の第6の態様にかかる透かし取出装置は、第1の態様から第5の態様のいずれかに記載の透かし挿入装置が前記透かし情報および前記透かし検証コードを挿入したプログラムを入力するプログラム入力手段と、前記プログラムから前記透かし情報を取り出し、前記透かし情報に基づいて配布先を一意に特定するID情報を生成する透かし検出手段と、を具備し、生成した前記ID情報に基づき前記配布先を特定するものである。

【0026】

これにより、不正にプログラムを流通した配布先を特定することができる。

【0027】

本発明の第7の態様にかかるプログラム不正配布防止システムは、第1の態様から第5の態様のいずれかに記載の透かし挿入装置と、第6の態様に記載の透かし取出装置と、を具備した構成を採る。

【0028】

このように第1の態様から第5の態様のいずれかに記載の透かし挿入装置と、第6の態様に記載の透かし取出装置と、を具備することにより、確実にプログラムの不正配布を防止できる。

【0029】

本発明の第8の態様は、第7の態様にかかるプログラム不正配布防止システムにおいて、前記透かし挿入装置を前記配布先に設けたものである。

【0030】

これにより、配布先に対して容易にプログラムを配布し、配布先において透かしを挿入するようにできる。このような形態は、単純にプログラムのみを配布することが好ましいシステムに効果的である。

【0031】

本発明の第9の態様は、配布先ごとに異なる透かし情報を前記プログラムに挿入するステップと、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させないものであり、前記配布先に関わらず同じ透かし検証コードを前記プログラムに挿入するステップと、を具備したことを特徴とする透かし挿入方法である。

【0032】

本発明の第10の態様は、プログラムの配布先ごとに異なる透かし情報をプログラムに挿入するステップと、前記透かし情報の挿入箇所の周辺もしくは前記プログラムの全体の仕様を維持したまま変換するステップと、を有することを特徴とする透かし挿入方法である。

【0033】

これにより、プログラムの差分をとったときに、透かし以外の部分が差分として検出されるため、差分攻撃に基づいて透かし挿入位置を特定することが困難になる。この結果、透かし改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【0034】

本発明の第11の態様は、コンピュータに、プログラムの配布先ごとに異なる透かし情報を前記プログラムに挿入するステップと、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記配布用プログラムを正しく動作させないもので

あり、前記配布先に関わらず同じ透かし検証コードを前記配布用プログラムに挿入するステップと、を行わせることを特徴とする透かし挿入プログラムである。

【0035】

本発明の第12の態様にかかる透かし挿入装置は、プログラムの配布先ごとに異なる透かし情報をプログラムに挿入する透かし挿入手段と、前記透かし情報を挿入する箇所以外の部分を前記プログラムの仕様を維持したまま変換する変換手段と、を具備した構成を採る。

【0036】

これにより、プログラムの差分をとったときに、透かし情報以外の部分が差分として検出されるため、差分攻撃に基づいて透かし挿入位置を特定することが困難になる。この結果、透かし改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【0037】

本発明の第13の態様は、第12の態様にかかる透かし挿入装置において、前記変換手段は、前記透かし情報を挿入する箇所以外の部分に、仕様に影響を与えない実行コードの組を挿入する。

【0038】

このように、仕様に影響を与えない実行コードの組を挿入することにより、差分攻撃が行われた際に透かし情報以外のコードが異なった箇所として検出されてしまうので、差分攻撃に基づく透かし改変、削除を確実に防ぐことができる。

【0039】

本発明の第14の態様は、第12の態様または第13の態様にかかる透かし挿入装置において、前記透かし情報の挿入箇所を示す識別情報を記憶する。

【0040】

これにより、識別情報を用いて透かし情報の挿入個所を容易に特定でき、透かし情報を容易に検出できる。

【0041】

本発明の第15の態様は、第14の態様にかかる透かし挿入装置において、前記識別情報は、メソッド名もしくは行番号である。

【0042】

これにより、識別情報により透かし情報の挿入個所を確実に検出できる。

【0043】

本発明の第16の態様は、第12の態様にかかる透かし挿入装置において、前記変換手段は、前記透かし情報を挿入する箇所以外の部分に、仕様に影響を与えないように難読化処理する。

【0044】

これにより、差分攻撃により透かし情報以外の部分が検出される。これにより、差分攻撃に基づいて透かし挿入位置を特定することが困難になる。

【0045】

本発明の第17にかかる透かし取出装置は、第12の態様から第16の態様のいずれかに記載の透かし挿入装置が前記透かし情報を挿入したプログラムを入力するプログラム入力手段と、前記プログラムから前記透かし情報を取り出す透かし検出手段と、を具備し、取り出した前記透かし情報に基づき配布先を特定する。

【0046】

これにより、プログラムの配布先を特定することができ、プログラムの不正流通を防止できる。

【0047】

本発明の第18にかかる透かし取出装置は、第15の態様または第16の態様に記載の透かし挿入装置が前記透かし情報を挿入したプログラムを入力するプログラム入力手段と、前記識別情報を取得し、前記識別情報から透かし挿入箇所を特定し、特定した前記透かし挿入個所のみから前記透かし情報を取り出す透かし検出手段と、を具備し、取り出した

前記透かし情報に基づき配布先を特定する。

【0048】

これにより、識別情報を用いることにより、透かし情報を容易に取り出すことができ、この透かし情報からプログラムの配布先を特定することができ、プログラムの不正流通を防止できる。

【0049】

本発明の第19の態様は、コンピュータに、プログラムの配布先ごとに異なる透かし情報をプログラムに挿入するステップと、前記透かし情報を挿入する箇所以外の部分を前記プログラムの仕様を変更することなく変換するステップと、行わせることを特徴とするプログラムである。

【0050】

本発明の第20の態様は、第12の態様にかかる透かし挿入装置において、前記変換手段は、前記透かし情報を挿入する箇所以外の部分であって、順序を入れ替えても仕様に影響を与えない部分の順序を変換する。

【0051】

これにより、差分攻撃によりプログラムの仕様に影響する部分が検出される。この結果、透かしの改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【0052】

本発明の第21の態様は、第20の態様にかかる透かし挿入装置において、前記仕様に影響を与えない部分を変換した履歴情報を保持し、前記履歴情報を用いて、前記仕様に影響を与えない部分の変換を配布先ごとに異なるようにする。

【0053】

これにより、仕様に影響を与えない部分の順序を、配布先ごとに確実に、かつ簡単に異なるようにできる。

【0054】

(実施の形態1)

本発明の実施の形態1にかかる透かし挿入装置および透かし取出装置を具備したプログラム不正配布防止システムについて添付図面を用いて説明する。

【0055】

図1は、実施の形態1にかかる透かし挿入による不正配布防止システムの構成図である。

【0056】

まず、配布元10は、プログラム配布の際に、透かし挿入装置20により、配布先40a、40bごとに、異なる透かしを挿入して配布する（配布先の2次配布を認めないものとする）。

【0057】

このように透かしの埋め込んで配布することで、不正な2次配布などプログラムが流出した際に、配布元10は、透かし取出装置30を用いて、流出先50に流出したプログラムから透かしを取り出して配布先を確認し、流出元（配布先）40a、40bを特定することができる。

【0058】

また、配布先40a、40bは、透かしによる流出元の特定を恐れて、不正な2次配布を控えることになる。

【0059】

このようにして、不正配布防止システムは、透かしによる不正配布を抑止する。

【0060】

次に、実施の形態1にかかる透かし挿入装置20について図2を用いて説明する。図2は、実施の形態1の透かし挿入装置の構成図である。

【0061】

透かし挿入装置20には、プログラム入力部201が設けられている。プログラム入力

部 201 は、透かしを入力するプログラムコードを入力する手段である。プログラム入力部 201 は、プログラムコードを透かし挿入部 202 に出力する。

【0062】

透かし挿入部 202 は、ID 情報生成部 205 により生成される ID 情報からプログラムに実際に埋め込む透かしを生成し、プログラム入力部 201 から出力されたプログラムコードに対し、透かしを入力する手段である。また、透かし挿入部 202 は、プログラム入力部 201 が出力したプログラムコードがソースコードであれば、ソースコードをコンパイルし、透かしの入力箇所をアセンブラコードの行番号として透かし用情報記憶部 206 に渡す。

【0063】

プログラム出力部 203 は、透かし挿入部 202 が透かしを入力したプログラムコードを出力する手段である。

【0064】

透かし用データ入力部 204 は、透かし用データを入力する。入力する透かし用データは、配布先を一意に特定する情報であり、配布先の住所、電話番号、会社名、氏名、電子メールアドレスなどである。また、透かし用データに、配布元の情報を入力してもよい。

【0065】

ID 情報生成部 205 は、透かし用データ入力部 204 により入力された透かし用データから一意に決定できる ID 情報を生成する。ID 情報は、入力したデータそのものであってもよいし、それを暗号化したデータであってもよい。また、ID 情報は、透かし用データを保存するデータベース上において透かし用データを一意に特定するための ID であってもよい。

【0066】

なお、本発明の実施の形態においては、ID 情報に基づいて透かし情報を生成する形態となっているが、必ずしも ID 情報に基づいて透かし情報を生成する必要はなく、透かし情報から一意に配布先を特定可能となっていれば良い。例えば、ソフトウェアに 1～N シーケンス番号を透かし情報として挿入し、配布先 A にシーケンス番号 i のソフトを配布、配布先 B にシーケンス番号 j のソフトを配布といったように透かし情報と配布先を一意に特定可能としてもよい。

【0067】

透かし用情報記憶部 206 は、透かし挿入部 202 が挿入した透かしの挿入箇所を記憶する手段である。具体的には、透かしを挿入したコードのアセンブラコード行番号を記憶する。

【0068】

次に、実施の形態 1 にかかる透かし取出装置 30 について図 3 を用いて説明する。図 3 は、実施の形態 1 における、透かし取出装置 30 の構成図である。

【0069】

プログラム入力部 301 は、透かしを挿入したプログラムを入力する手段である。

【0070】

透かし検出部 302 は、プログラム入力部 301 から出力されたプログラムを逆アセンブリングし、透かし用情報記憶部 305 より得られる透かし挿入箇所（アセンブラコード行番号）から入力された透かしを取り出す。そして、透かし検出部 302 は、取り出した透かしから ID 情報を生成し、ID 情報記憶部 304 に渡す。

【0071】

ID 情報記憶部 304 は、透かし検出部 302 より得られる ID 情報から、配布先の情報を生成する手段である。ID 情報記憶部 304 は、ID 情報がデータベースのデータの ID である場合には、ID からデータを取り出すことで、配布先の情報を取得する。また、ID 情報記憶部 304 は、ID 情報が配布先の情報の暗号化データである場合には、復号して配布先の情報を取得する。

【0072】

透かし用情報記憶部 305 は、配布したプログラムの透かし挿入箇所を記憶している手段である。透かし挿入箇所の情報は、透かし挿入装置 20 の透かし用情報記憶部 206 より得る。

【0073】

出力部 303 は、取得された配布先の情報を出力する手段である。

【0074】

次に、実施の形態 1 にかかる透かし挿入部 202 の透かし挿入動作について図 4 を用いて説明する。図 4 は、実施の形態 1 にかかる透かし挿入部 202 の動作を表すフローチャートである。

【0075】

まず、透かし挿入部 202 は、配布先 40 の情報から生成される ID 情報 I から、実際にプログラムに挿入する透かし情報 X1、X2 を生成関数 F(1) により生成する (ステップ 401)。

【0076】

続いて、透かし挿入部 202 は、透かし情報 X1、X2 を入力としたとき、定数 C1 を出力する関数 F21 および定数 C2 を出力する関数 F22 を構成する (ステップ 402)。

。

【0077】

続いて、透かし挿入部 202 は、透かし情報 X1、X2 を変数 val1, val2 に代入する式をプログラムコード中に埋め込む (ステップ 403)。

【0078】

続いて、透かし挿入部 202 は、プログラムコード中に、F21(val1, val2) を変数 val3 に、F22(val1, val2) を変数 val4 に代入する式を埋め込む (ステップ 404)。

【0079】

続いて、透かし挿入部 202 は、透かし検証コードである、変数 val3 と定数 C1 が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐と、変数 val4 と定数 C2 が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐と、をプログラムコード中に埋め込む (ステップ 405)。

【0080】

そして、透かし挿入部 202 は、ステップ 403 からステップ 405 において透かし情報および透かし検証コードを挿入した箇所を透かし用情報記憶部 206 に記憶する (ステップ 406)。

【0081】

このようにして、透かし挿入部 202 は、プログラムに透かしである透かし情報および透かし検証コードを挿入する。

【0082】

なお、透かし挿入部 202 は、ステップ 403 からステップ 405 において、挿入した式および条件分岐 (透かし検証コード) を、プログラムの実行順に入力する。ただし、F1 は、X1、X2 より I を一意に生成する F1 の逆関数を持つことを条件とし、F21、F22 は、 $F21(X1, X2) == C1$ かつ $F22(X1, X2) == C2$ を X1、X2 以外の時には満たさないことを条件とする (“==” は値が等しいことをあらわす。)。

。

【0083】

たとえば、ID 情報 I = 12345678、F1 は 8 桁の値を 4 桁目から 2 つの値に分割する関数、 $F21(x, y)$ 、 $F22(x, y)$ は $ax + by$ という 2 変数一次関数、 $C1 = 2345$ 、 $C2 = 5678$ の場合を考える。

【0084】

この場合、まず、F1 から、透かし情報 $X1 = 1234$ 、 $X2 = 5678$ が生成される。また、F21、F22 は、 $a1 \times 1234 + b1 \times 5678 = 2345$ 、 $a2 \times 123$

$4 + b_2 \times 5678 = 5678$ を満たす a_1 , a_2 , b_1 , b_2 を求めることにより構成する。たとえば、 $a_1 = 1$, $a_2 = 0$. 195667, $a_2 = 3$. 700972, $b_2 = 0$. 195667は条件を満たす。

【0085】

次に、実施の形態1を適用した場合に生成されるプログラムの例を、図5に示す。

【0086】

500aは、プログラム入力部201が入力した基本となる基本プログラムである。プログラム500b、500cは、基本プログラム500aに透かしである透かし情報および透かし検証コードを入力した透かし挿入プログラムである。

【0087】

まず、透かし挿入部202は、ステップ403において、透かし挿入プログラム500b、500cに、それぞれ別のID情報Ia(12345678)、Ib(11112222)より生成された透かし情報X1a(1234)、X1b(5678)とX2a(1111)、X2b(2222)を入力する(図中501に示す部分)。

【0088】

次に、透かし挿入部202は、ステップ404において、透かし挿入プログラム500b、500cに、それぞれ別のF21、F22を透かし挿入プログラム500b、500cに挿入する(図中502に示す部分)。

【0089】

そして、透かし挿入部202は、ステップ405において、透かし検証コードとして、変数val3と定数C1(2345)が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐(assert(0))と、変数val4と定数C2(5678)が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐(assert(0))と、をプログラムコード中に埋め込む(図中503に示す部分)。

【0090】

ここで、着目すべき点は、2つのプログラム500bと500cの差分をとると、透かし情報である501、502の部分は検出されるが、透かし検証コードである条件分岐503は検出されないことである。これにより、プログラム500b、500cに差分攻撃をすることにより、透かし入力箇所を検出し、検出部分の改ざん、削除を行ったとしても、透かし検証コードである条件分岐503の部分の改ざん、削除は行えない。よって、透かし検証コード503の部分が条件と合わなくなり、プログラムが動作しなくなる。

【0091】

このように、差分攻撃により検出された箇所だけを改変、削除するという単純な手法では、全ての透かしを削除した正常に動作するプログラムを入手できなくさせることが可能となる。

【0092】

なお、わかりやすさのため、図5ではソースコードを用いて説明しているが、バイナリコードの場合でも同じことが言える。また、図5では、条件分岐503は、条件文が真であった場合にプログラムを停止するよう処理しているが、そうでなく、プログラムが異常な動作をするように(たとえば、a++とするなど)プログラム中の変数値を変更するように処理することもできる。

【0093】

さらに、実施の形態1では、ID情報から、2つの透かし情報を生成しているが、3つ以上の透かし情報を生成することとしてもよい。

【0094】

次に、実施の形態1にかかる透かし検出部302の動作について図6を用いて説明する。図6は、実施の形態1の透かし検出部302の動作を表すフローチャートである。

【0095】

まず、透かし検出部302は、プログラムの実行コードを逆アセンブルする(ステップ1001)。

【0096】

その後、透かし検出部302は、透かし用情報記憶部305を参照し、プログラムへの透かし挿入箇所を記憶した記憶情報（挿入箇所を示すアセンブラ行番号）を取得し、これに基づいて透かし情報X1、X2の入力箇所を特定する。そして、透かし検出部302は、透かし情報X1、X2をプログラムから取り出す（ステップ1002）。

【0097】

続いて、透かし検出部302は、透かし情報X1、X2を生成する際に使用した関数F1の逆関数を使用して、ID情報Iを生成する（ステップ1003）。

【0098】

このようにして、透かし検出部302は、ID情報Iを取得して、配布先40の特定を行う。

【0099】

なお、上記の方法では、実行コードの配布先もしくは流出先で、最適化、難読化といった処理によりコードの実行順序を入れ替えられた場合に、透かし情報の入力箇所のアセンブラ行番号が変化してしまい、透かし情報を得られない可能性がある。このような場合を考慮して、ステップ1002の処理を、挿入箇所を示すアセンブラ行番号の周辺の行において、代入命令をさがし、代入命令のオペランド部を取り出すという処理に変更してもよい。

【0100】

以上説明したように、実施の形態1によれば、透かし検証コード（図5の503の部分）は、配布先に関わらず同じなので、差分攻撃により、透かし検証コード（図5の503の部分）が差分として検出されないようにできる。これにより、差分攻撃により透かし検証コードの挿入位置を検出することができない。この結果、差分攻撃により検出された箇所だけを改変、削除するという単純な手法では、全ての透かしを改変、削除できなくなり、正常に動作する透かしのない（もしくは改変された）プログラムを生成することが不可能になる。よって、配布先は、透かしがなく、かつ正常に動作するプログラムを生成できないので、プログラムを不正に流通させることができなくなる。

【0101】

なお、透かし挿入装置20および透かし取出装置30の行う処理をプログラムにし、汎用のコンピュータに実行させる形態であってもよい。

【0102】

（実施の形態2）

本発明の実施の形態2は、不正にプログラムを配信しようとした者が、差分攻撃により透かし情報を検出し、検出した透かし情報に用いられている関数により生成される変数を使用している箇所（図5に示す503の部分）を検出し、検出した箇所を改変、削除することにより、実施の形態1の透かし検証コードを改変、削除しようとした場合に対応したものである。

【0103】

具体的には、透かし情報を用いたものであり、かつプログラムを正しく動作させるのに必要な透かし検証コードをプログラム中に挿入するようにしたものである。

【0104】

これにより、上述した手順により透かし情報を用いた透かし検証コードを検出し、改変、削除した場合には、プログラムを正常に動作させなくすることができる。

【0105】

以下、実施の形態2について詳細に説明する。実施の形態2における透かし挿入装置と、実施の形態1における透かし挿入装置20との違いは、透かし挿入部202の動作である。

【0106】

次に、実施の形態2の透かし挿入部の動作について図7を用いて説明する。図7は、実施の形態2の透かし挿入部の動作のフローチャートである。

【0107】

ステップ601およびステップ602の動作は、実施の形態1で説明した図4のステップ401およびステップ402の動作と同様であるので説明を省略する。

【0108】

続いて、透かし挿入部は、透かし情報X1、X2より、 $C1 + C2 + C3 = 0$ となるC3を生成する関数F3を生成する（ステップ603）。

【0109】

続いて、透かし挿入部は、透かし情報X1、X2を変数val1、val2に代入する式をプログラムコード中に埋め込む（ステップ604）。

【0110】

続いて、透かし挿入部は、プログラムコード中に、F21(val1, val2)を変数val3に、F22(val1, val2)を変数val4に代入する式を埋め込む（ステップ605）。

【0111】

続いて、透かし挿入部は、透かし検証コードとして、変数val3と定数C1が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐と、変数val4と定数C2が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐と、をプログラムコード中に埋め込む（ステップ606）。

【0112】

続いて、透かし挿入部は、F3(val1, val2)を変数val5に代入する式を埋め込む（ステップ607）。

【0113】

次に、透かし挿入部は、透かし検証コードとして、 $val3 + val4 + val5$ をオリジナルコードの0を判定する判定文に加算するコードをプログラム中に挿入する（ステップ608）。

【0114】

そして、透かし挿入部202は、ステップ604からステップ608において透かし情報および透かし検証コードを挿入した箇所を透かし情報記憶部206に記憶する（ステップ609）。

【0115】

このようにして、透かし挿入部は、プログラムに透かしを挿入する。

【0116】

ここで、着目すべき点は、ステップ608で挿入される $val3 + val4 + val5$ には、差分検出で検出される変数、val3、val4、val5が含まれている点であり、 $val3 + val4 + val5$ がプログラムの動作に関わる判定文の0の部分に挿入されている点である。これにより、不正使用者が、差分攻撃により変数(val3、val4、val5)を検出し、検出した変数を用いている関数により生成される変数を使用している箇所を改変、削除しようとした場合に、プログラムの動作にかかわる判定文も改変、削除してしまうことになる。よって、プログラムが正常に動作しなくなり、不正使用もできなくなる。

【0117】

次に、実施の形態2にかかる透かし挿入部が生成したプログラムについて図8を用いて説明する。

【0118】

800aは、プログラム入力部201が入力した基本となる基本プログラムである。プログラム800bは、基本プログラム800aに透かしを入力した透かし挿入プログラムである。

【0119】

プログラム800bには、ステップ604において、701で示される部分に透かし情報が挿入され、ステップ605、ステップ607において、702で示される部分に透かし

し検証用の計算式（コード）が挿入される。

【0120】

そして、プログラム800bには、703で示される部分に、ステップ608の処理結果が挿入される。また、プログラム800bには、ステップ606において、704で示される部分に透かし検証コードが挿入される。

【0121】

このようにプログラム800bを生成することにより、不正に使用しようとした者が、差分攻撃によりプログラム800bから透かし検証コード703を検出し、透かし検証コードを改変、削除すると、透かし検証コード703は、仕様（オリジナルの祖コードにおける、プログラムの入出力関係）に関連するコードであるので、このコードを削除するとプログラムが正しく動作しなくなる。

【0122】

透かしのうち、透かし検証コード703の判定文のみ変更しようとするためにはプログラムの仕様を理解して、透かし検証コード703が仕様に関連あるコードであることを知る必要がある。これは、プログラムの構造を理解した物が、時間をかけて行う必要があり、機械的な処理により透かしの削除はできない。

【0123】

なお、 $C1 + C2 + F3 = 0$ でなくても良い。ただし、この場合は、 $C1 + C2 + F3$ により得られる値を用いた判定文に $C1 + C2 + F3$ を挿入すればよい。例えば、 $C1 + C2 + F3 = 1$ である場合は、1を判定する判定文の1を $C1 + C2 + F3$ を置換する。

【0124】

以上説明したように、実施の形態2によれば、差分攻撃により検出した透かし情報（701、702）に用いられている関数により生成される変数を使用している箇所（図8に示す703の部分）を検出し、改変、削除した場合に、プログラムが正常に動作しないようになる。つまり、正常に動作する透かしのない（もしくは改変された）プログラムを生成することを不可能にすることができるので、プログラムの不正配布を防止できる。

【0125】

（実施の形態3）

本発明の実施の形態3は、透かし情報および透かし検証コードを入力した箇所周辺のコード、もしくはコード全体を難読化などの処理をすることにより改変するものである。これにより、差分攻撃により透かし以外のコードが検出されてしまうので、差分攻撃に基づく透かし改変、削除を確実に防ぐことができる。

【0126】

以下、実施の形態3について詳細に説明する。実施の形態3における透かし挿入装置と、実施の形態1における透かし挿入装置20との違いは、透かし挿入部202の動作である。

【0127】

次に、実施の形態3の透かし挿入部202の動作について図9を用いて説明する。図9は、実施の形態3の透かし挿入部202の動作フローチャートである。

【0128】

まず、透かし挿入部202は、変数*i*に初期値1を代入する（ステップ800）。次に、透かし挿入部は、ID情報を*n*個の情報に分割して、透かし情報X（1）、X（2）…X（*n*）を生成する（ステップ801）。

【0129】

続いて、透かし挿入部202は、プログラムソースコード中のループ部（*while*、*for*文）を検出し（ステップ802）、ループの内部に透かし情報X（*i*）を挿入する（ステップ803）。

【0130】

その後、透かし挿入部202は、“ループを含むプログラムを難読化する方法の提案”（門田ら、信学論D-I， Vol. J80-D-I， No. 7， pp. 644-

652, July 1997)に記載される方法を適用することで、挿入箇所のループ部を難読化する(ステップ804)。この際、プログラムの難読化の仕方に複数のバリエーションがあるが、バリエーションをランダム(もしくは過去に配布したプログラムに施した難読化と重ならないよう)に選択する。

【0131】

そして、透かし挿入部202は、変数*i*が透かし情報の数*n*以下か判断し(ステップ805)、変数*i*が*n*以下の場合変数*i*をインクリメントし(ステップ806)、ステップ802の処理に以降する。一方、ステップ805において、変数*i*が*n*以下でないと判断した場合、つまり全ての透かし情報を入力した場合は、透かし挿入部202は、その後ソースコードをコンパイルし、透かし情報が入力されたアセンブラコードの行番号を記憶してプログラムを出力し、処理を終了する(ステップ807)。

【0132】

次に、実施の形態3にかかる透かし挿入部202が生成したプログラムについて図10を用いて説明する。900aは、プログラム入力部201が入力した基本となる基本プログラムである。プログラム900b、900cは、基本プログラム900aに透かし情報901を入力した透かし挿入プログラムである。

【0133】

プログラム900b、900cには、難読化により、実装は異なるものの、仕様(プログラムの入出力の関係)は変化していない。プログラム900b、900cの差分をとると、透かし以外の場所もプログラムコードが変化しているため、透かし以外の部分902a、902bも差分として検出される。

【0134】

したがって、プログラム900b、900cの透かしを改変、削除するためには、プログラムを解析し、どの部分がプログラムの仕様に関係のない透かしとなっているかを探し出す必要がある。プログラムの仕様に関係ない部分であるかどうかを判定するためには、プログラムの仕様を理解する必要があるため、この方法により埋め込まれた透かしを機械的に削除することは困難となる。

【0135】

以上説明したように、実施の形態3は、透かし挿入部が、プログラムの透かしを挿入する箇所以外の部分に、プログラムの仕様に影響を与えないように難読化処理をする改変手段としての動作もするので、差分攻撃により透かし以外のコードであるプログラムの仕様に関係する部分が検出される。これにより、差分攻撃に基づいて透かし挿入位置を特定することが困難になる。この結果、透かし改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【0136】

(実施の形態4)

本発明の実施の形態4は、配布先に透かし挿入装置を保持し、配布したプログラムに対して配布先で透かしを付与するものである。

【0137】

以下、実施の形態4にかかる不正配布防止システムの構成について図11を用いて説明する。図11は、実施の形態4にかかる透かし挿入による不正配布防止システムの構成図である。なお、すでに説明した部分と同一の部分には同一の符号を付与してある。

【0138】

本システムでは、まず、配布元1100が、配布先1110、1120に対して、それぞれ配布先1110、1120を一意に決定するID情報1101、1102を配布する。

【0139】

これに対して、配布先1110、1120の透かし挿入装置20a、20bでは、ID情報1101、1102を記憶しておく。

【0140】

続いて、配布元 1100 は、配布先 1110、1120 にプログラム 1103 を配布する。

【0141】

これに対して、配布先 1110、1120 では、配布されたプログラム 1103 に対して、透かし挿入装置 20a、20b を利用して透かしを挿入したプログラム 1111、1121 を生成する。

【0142】

なお、透かし挿入装置 20a、20b は、実施の形態 1 から実施の形態 3 のいずれかにかかるものであってもよい。

【0143】

その後、透かし挿入装置 20a、20b は、透かしを挿入した箇所を記憶した記憶情報 1104、1105 を配布元 1100 に送信し、配布元 1100 では記憶情報 1104、1105 を保存する。

【0144】

配布元 1100 は、配布先 1110 が、流出先 1130 に不正な 2 次配布を行った場合には、流出したプログラム 1112 を取得し、記憶情報 1104、1105 とともに透かし取出装置 30 に入力する。そして、配布元 1100 は、透かし取出装置 30 において、配布先 1110、1120 を特定する ID 情報 1107 を入手する。そして、配布元 1100 は、配布先 1110、1120 に配布した ID 情報 1101、1102 と、入手した ID 情報 1107 を比較し、不正にプログラムを流出した配布先 1110、1120 を特定する。

【0145】

以上説明したように、実施の形態 4 によれば、不特定多数の配布先に対して容易にプログラムを配布し、配布先において透かしを挿入するようにできる。このような形態は、単純にプログラムのみを配布することが好ましいシステム、たとえばデジタル放送を利用したプログラムの配布であるとか、IP ネットワークでマルチキャスト、ブロードキャストを利用したプログラム配布などに適用すると効果的である。

【0146】

(実施の形態 5)

本発明の実施の形態 5 は、透かしを挿入するメソッドやその他のメソッドが実装されている箇所に、プログラムの仕様に影響を与えない偽装コードを追加することにより、プログラムを改変するものである。これにより、差分攻撃が行われた際に透かし以外のコードが異なった箇所として検出されてしまうので、差分攻撃に基づく透かし改変、削除を確実に防ぐことができる。

【0147】

次に、実施の形態 5 にかかる透かし挿入装置 1200 について図 12 を用いて説明する。図 12 は、実施の形態 5 の透かし挿入装置の構成図である。

【0148】

実施の形態 5 にかかる透かし挿入装置 1200 のプログラム入力部 201 は、他の実施の形態における透かし挿入装置 20 のプログラム入力部 201 と同じ動作をする。

【0149】

透かし挿入装置 1200 には、プログラム入力部 201 により出力されたプログラムの実行に影響を与えない余分なダミーメソッドを入力するダミーメソッド入力部 1203 が設けられている。ダミーメソッド入力部 1203 は、入力したダミーメソッドをダミーメソッド挿入部 1201 に出力する。

【0150】

ダミーメソッド挿入部 1201 は、ダミーメソッド入力部 1203 で入力されたダミーメソッドを、透かしを埋め込むための領域として追加する手段である。ダミーメソッド挿入部 1201 は、ダミーメソッドの追加されたプログラムを偽装コード挿入部 1202 に出力する。

【0151】

偽装コード挿入部1202は、ダミーメソッド挿入部1201から出力されたプログラム全体のメソッド（ダミーメソッドを含む全てのメソッド）が実装されている箇所に、プログラムの実行に影響を与えず、プログラムの実行結果には必要としない偽装コードの組を挿入することにより、プログラムの仕様を変更することなく改変する改変手段である。挿入する偽装コードとしては、PUSHとPOPの組などが考えられる。

【0152】

透かし挿入部202、プログラム出力部203、透かし用データ入力部204、およびID情報生成部205は、他の実施の形態における透かし挿入装置20の透かし挿入部202とプログラム出力部203、透かし用データ入力部204、およびID情報生成部205と、それぞれ同じ手段である。

【0153】

透かし用情報記憶部1204は、透かし挿入部202が挿入した透かしに対して、透かしとして用いた文字や数値、および記号とビット列との対応情報と、ビット列と命令コードとの対応情報を記憶する。また、透かし用情報記憶部1204は、透かしを挿入したダミーメソッドの識別情報として、メソッド名や行番号を保存する。さらに、透かし用情報記憶部1204は、透かし用データとして暗号化したデータを使用した場合には、データを復号するための鍵の情報をあわせて記憶する。

【0154】

これにより、識別情報を用いて、透かしの挿入個所を容易に特定でき、透かし情報を容易に検出できる。

【0155】

次に実施の形態5にかかる透かし取出装置30について説明する。実施の形態5における透かし取出装置30と、他の実施の形態における透かし取出装置30との違いは、透かし検出部302、透かし用情報記憶部305の動作である。

【0156】

透かし検出部302は、プログラム入力部301から出力されたプログラム中で、透かし用情報記憶部305から得られる、透かしを挿入したメソッドの識別情報を獲得し、識別情報の表すメソッドを検査する。

【0157】

次に、透かし検出部302は、同じ透かし用情報記憶部305から得られる、透かしとして用いた文字や数値、および記号とビット列との対応と、ビット列と命令コードとの対応を利用して、命令コードからビット列、ビット列から文字や数値、および記号へ変換することにより、プログラムに挿入されている透かし情報を取り出す。

【0158】

透かし検出部302は、取り出した透かしからID情報を生成し、ID情報記憶部304に出力する。

【0159】

透かし用情報記憶部305は、透かしの挿入されているメソッドの識別情報を保持している手段である。また、透かし用情報記憶部305は、配布したプログラムの透かしとして用いた文字や数値、および記号とビット列との対応と、ビット列と命令コードとの対応も記憶している。さらに、透かし用情報記憶部305は、挿入された透かしのデータが暗号であった場合に、暗号を復号するための鍵も保持している。透かし用情報記憶部305は、文字や数値、および記号とビット列との対応と、ビット列と命令コードとの対応、透かしの挿入されているメソッドの識別情報、暗号データを復号するための鍵は、透かし挿入装置1200の透かし用情報記憶部1204より得る。

【0160】

次に、実施の形態5の偽装コード挿入部1202と透かし挿入部202の動作について図13を用いて説明する。図13は、実施の形態5の偽装コード挿入部1202と透かし挿入部202の動作フローチャートである。

【0161】

まず、偽装コード挿入部1202は、変数*i*に初期値1を代入する（ステップ1300）。次に、透かし挿入部202は、文字や数値、および記号とビット列の対応を用いて、ID情報より透かし情報*S*を生成する（ステップ1301）。

【0162】

続いて、偽装コード挿入部1202は、プログラム中で、メソッドが実装されている箇所であるメソッド部を検出し（ステップ1302）、変数*i*がプログラム中の総メソッド数以下であるかの判断を行い（ステップ1303）、変数*i*が総メソッド数以下である場合には、プログラムの仕様に影響を与えない、本来は必要ではない偽装コードを挿入する（ステップ1304）。

【0163】

このとき挿入する偽装コードは、複数のバリエーションがあるが、バリエーションをランダム、もしくは過去に配布したプログラムに挿入した偽装コードと重複しないように選択する。つまり、差分攻撃により、偽装コードが抽出されるように偽装コードを挿入する。

【0164】

次に、透かし挿入部202は、検出されたメソッド部がダミーメソッドであるかを判断し（ステップ1305）、ダミーメソッドである場合には、“プログラムに電子透かしを挿入する一手法”、（門田ら、1998年暗号と情報セキュリティシンポジウム、SCIS'98-9.2. A, Jan. 1998）を適用することで、透かし情報*S*を挿入する（ステップ1306）。

【0165】

また、このとき、透かし挿入部202は、ダミーメソッドの識別情報を保存する（ステップ1307）。

【0166】

そして、透かし挿入部202は、変数*i*をインクリメントし（ステップ1308）、ステップ1302の処理に移行する。

【0167】

一方、ステップ1303において、変数*i*がプログラムコードの総メソッド数以下でないと判断した場合、つまり全てのメソッドに偽装コードを挿入し、そのうちのダミーメソッドに透かし情報を挿入した場合は、透かし挿入部202は、透かし情報が埋め込まれたプログラムを出力する（ステップ1309）。

【0168】

実施の形態5にかかる透かし挿入部202が生成したプログラムは、偽装コードの挿入により、実装は異なるものの、仕様（プログラムの入出力の関係）は変化していない。また、それぞれのプログラムで異なる偽装コードが挿入されているため、透かしの挿入メソッドを特定するためにプログラム間の差分をとると、透かしが挿入されているメソッド以外のメソッドも差分として検出される。

【0169】

したがって、プログラムの透かしを改変、削除するためには、プログラムを解析し、どのメソッドがプログラムの仕様に関係のない透かし挿入用のダミーメソッドとなっているかを探し出す必要がある。プログラムの仕様に関係ない部分であるかどうかを判定するためには、プログラムの仕様を理解する必要があるため、この方法により埋め込まれた透かしを機械的に削除することは困難となる。

【0170】

次に、実施の形態5を適用した場合に生成されるプログラムの例を、図14に示す。

【0171】

図中1600aで示されるプログラムは、基本となるソースプログラムである。このプログラム1600aをコンパイルしたものが、プログラム入力部201より透かし挿入装置1200に入力されるプログラムであるが、ここでは、説明の管理化のために、コンパ

イルしたプログラム 1600a を逆アセンブルしたプログラム 1600b を用いて説明を行う。

【0172】

また、プログラム 1600c とプログラム 1600d は、それぞれ異なる透かしと偽装コードを挿入したプログラムである。またそれぞれのプログラム 1600a ~ 1600d において、A2 のメソッドがダミーメソッドを表し、命令ニーモニックの前にある数字はメソッドごとの行番号を示している。

【0173】

まず、透かし挿入部 202 は、ステップ 1301 において、透かし挿入プログラム 1600c、1600d 用に、それぞれ別の ID 情報 I1 ((C) 01)、I2 ((C) 02) より、1文字6ビットで透かし情報 S1 (100111 001101 101000 000000 000001)、S2 (100111 001101 101000 000 000010) を生成する。

【0174】

次に、偽装コード挿入部 1202 は、ステップ 1302 において、透かし挿入プログラム 1600b で、メソッド部を検出し、ステップ 1304 においてダミーメソッドではない A1 にそれぞれ異なる偽装コードを挿入する (図 14 中の 1601 で示される部分)。

【0175】

さらに、透かし挿入部 202 は、メソッドがダミーメソッド A2 の場合には、ステップ 1306 において、透かし挿入プログラム 1600b に対して、透かし情報 S1 と S2 から埋め込み対象の命令に割り当てられたビット数分だけ透かし情報として埋め込む。

【0176】

この例では、プログラム 1600b のメソッド A2 内の `iconst__0` が埋め込み対象命令で、2ビットの情報量が割り当てられており、S1 と S2 から 2ビットが取り出されて埋め込みが行われる (図 14 の 1602 で示される部分)。

【0177】

このとき、透かし挿入部 202 は、各文字の下位ビットから取り出しを行い、一文字分全て取り出し終わった場合には、次の文字の下位ビットから取り出しを行う。

【0178】

また、偽装コード挿入部 1202 は、メソッド A2 に対しても、メソッド A1 と同じように偽装コードの挿入も行う (図 14 の 1603 で示される部分)。

【0179】

もし、プログラム 1600c と 1600d の配布先が結託し、透かし情報の挿入場所を特定するために、プログラム間の差分をとったとしても、透かし情報ではない 1601 で示される部分や 1603 で示される部分が、透かし情報の 1602 とともに検出されてしまうため、差分攻撃に基づいて透かし情報の挿入位置を特定することが困難となる。

【0180】

したがって、透かし情報の機械的な改変、削除を確実に防ぐことが可能で、プログラムの不正流通を防止することができる。

【0181】

次に、実施の形態 5 にかかる透かし検出部 302 の動作について図 15 を用いて説明する。図 15 は、実施の形態 5 の透かし検出部 302 の動作を表すフローチャートである。

【0182】

まず、透かし検出部 302 は、透かし用情報記憶部 305 よりダミーメソッドの識別情報を獲得する (ステップ 1500)。

【0183】

続いて、透かし検出部 302 は、獲得した識別情報を用いて、プログラム中でダミーメソッドが実装されているダミーメソッド部とメソッド部を検出し (ステップ 1501)、透かし用情報記憶部 305 に記憶しておいたビット列と命令コードの対応を用いて、ダミーメソッド部から透かし情報 S を取り出す (ステップ 1502)。

【0184】

さらに、透かし検出部302は、ID情報記憶部304で記憶された情報と、取り出した透かし情報Sから、プログラムの配布先を一意に特定するID情報を生成し（ステップ1503）、ID情報を出力（ステップ1504）して終了する。

【0185】

このように、透かし検出部302は、識別情報を用いることにより、ダミーメソッド部とメソッド部を容易に検出し、ダミーメソッド部から透かし情報Sを取り出すことにより、プログラムの配布先を特定できる。この結果、プログラムの不正流通を防止できる。

【0186】

以上説明したように、実施の形態5によれば、プログラム中に透かし情報だけでなく、仕様に影響を与えない実行コードの組である偽装コードを挿入することができる。これにより、差分攻撃が行われた際に透かし以外のコードが異なった箇所として検出されてしまうので、差分攻撃に基づく透かし改変、削除を確実に防ぐことができる。

【0187】

(実施の形態6)

本発明の実施の形態6は、透かし情報の挿入箇所以外の一部もしくはプログラム全体のコードの順序を入れ替えることにより、プログラムを改変するものである。これにより、差分攻撃が行われた際に透かし以外のコードが異なった箇所として検出されるので、差分攻撃に基づく透かしの改変、削除を確実に防ぐことができる。

【0188】

以下、実施の形態6について詳細に説明する。実施の形態6における透かし挿入装置と、実施の形態1における透かし挿入装置20との違いは、透かし挿入部202の動作である。

【0189】

次に、実施の形態6の透かし挿入部の動作について図16を用いて説明する。図16は、実施の形態6の透かし挿入部の動作フローチャートである。

【0190】

まず、透かし挿入部202は、変数*i*に初期値1を代入する（ステップ1601）。次に、透かし挿入部202は、コード（プログラム）中に埋め込むための（配布先ごとに異なる）透かし情報SをID情報より生成する（ステップ1602）。

【0191】

続いて、透かし挿入部202は、コード全体で、順序を入れ替えても仕様に影響を与えない、つまり仕様を維持できるコード部分を抽出する（ステップ1603）。ここでコード部分とは、複数のコードからなるプログラムの一部をいう。

【0192】

そして、透かし挿入部202は、変数*i*が、順序を入れ替えても仕様を維持できるコード部分の数*N*以下であるか判断し（ステップ1604）、変数*i*が*N*以下であった場合には、そのコード部分に含まれるコードの順序を入れ替え（ステップ1605）、*i*をインクリメントして（ステップ1606）、ステップ1604に移行する。

【0193】

変数*i*が*N*以下でない場合には、透かし挿入部202は、透かし情報Sをコード中に挿入し（ステップ1607）、ソースコードをコンパイルし、透かし情報Sが挿入されたアセンブラコードの行番号を記憶してプログラムを出力し、処理を終了する（ステップ1608）。

【0194】

このように、透かし挿入部202は、順序を入れ替えても仕様を維持できるコード部分の順序を入れ替えることにより、透かし情報を挿入する個所以外の部分をプログラムの仕様を維持したまま変換する。

【0195】

次に、実施の形態6にかかる透かし挿入部が生成したプログラムについて図17を用い

て説明する。プログラム1700aは、プログラム入力部201により入力されるオリジナルのプログラムである。また、プログラム1700b、1700cは、オリジナルプログラム1700aに配布先ごとに異なる透かし1702b、1702cを挿入したプログラムである。

【0196】

また、プログラム1700b、プログラム1700cは、コード部分1701b、1701c、および1703b、1703cを含むプログラムである。コード部分1701b、1701c、および1703b、1703cは、オリジナルのプログラム1700aに含まれる透かしを挿入するコード部分でなく、かつコードを並び替えても仕様を維持できるコード部分1701a、1703aのコードの並びを変えたものである。

【0197】

このように、プログラム1700b及びプログラム1700cは、プログラム1700aに対してそれぞれ異なる命令順序に変換されているが、全体の仕様は変化していない。つまり、プログラム1700b及びプログラム1700cは、プログラム1700aは、仕様を維持したまま変換されている。ここでプログラム1700b、1700cの差分をとると、透かし以外の場所もプログラムコードが変化しているため、透かし以外の部分であるコード部分1701b、1701c、1703b、1703cの部分も差分として検出される。

【0198】

したがって、プログラム1700b、1700cの透かしを改変、削除するためには、プログラムを解析し、どの部分がプログラムの仕様に影響のない透かしとなっているかを探し出す必要がある。プログラムの仕様に影響のない部分であるかどうかを判定するためには、プログラムの仕様を理解する必要があるため、この方法により埋め込まれた透かしを機械的に削除することは困難となる。

【0199】

以上説明したように、実施の形態6は、透かし挿入部202が、プログラムの透かしを挿入する箇所以外のプログラム部分のうち、命令順序を入れ替えても仕様を維持できるプログラム部分を検出し、プログラムの仕様に影響を与えないように、つまり仕様を維持したまま、命令順序を入れ替えても良いプログラム部分の順序変換する変換手段としての動作をする。これにより、差分攻撃により透かし以外のコードであるプログラムの仕様に影響するプログラム部分が検出される。この結果、透かしの改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【0200】

また、実施の形態6によれば、命令順序を入れ替えても良いプログラム部分の順序変換を、プログラム部分中の命令文の順列を求め、配布先ごとに異なるように選択された順列に従い変換する。これにより、配布先ごとに命令順序を入れ替えても良いプログラム部分の命令順序が異なるようになる。よって、命令順序を入れ替えても良いプログラム部分の特定が困難になり、確実に透かしの改変、削除を確実に防ぐことができる。

【0201】

また、命令文の順列に従う変換以外の方法として、命令順序を入れ替えてもよいプログラム部分の順序をランダムに変換して、配布先ごとに異なるようにしてもよい。

【0202】

なお、順序を入れ替えてもよいコード部分に含まれるコードの順序変換の履歴情報を保持しておき、この履歴情報を用いて、順序を入れ替えてもよいコード部分の変換を配布先ごとに異なるようにしてもよい。

【0203】

これにより、順序を入れ替えてもよいコード部分に含まれるコードの順序変換を、配布先ごとに確実に、かつ簡単に異なるようにできる。

【産業上の利用可能性】

【0204】

以上説明したように、本発明によれば、透かしの挿入箇所を特定することが困難なように透かしを挿入できるので、ネットワークを利用したコンピュータプログラムの流通等に広範囲に適応できる。

【図面の簡単な説明】

【0205】

【図1】本発明の実施の形態1にかかる透かし挿入による不正配布防止システムの構成図

【図2】実施の形態1の透かし挿入装置の構成図

【図3】実施の形態1における、透かし取出装置の構成図

【図4】実施の形態1にかかる透かし挿入部の動作を表すフローチャート

【図5】実施の形態1を適用した場合に生成されるプログラムを示す図

【図6】実施の形態1の透かし検出部の動作を表すフローチャート

【図7】本発明の実施の形態2にかかる透かし挿入部の動作のフローチャート

【図8】実施の形態2にかかる透かし挿入部が生成したプログラムを示す図

【図9】本発明の実施の形態3にかかる透かし挿入部の動作フローチャート

【図10】実施の形態3にかかる透かし挿入部が生成したプログラムを示す図

【図11】本発明の実施の形態4にかかる透かし挿入による不正配布防止システムの構成図

【図12】本発明の実施の形態5における透かし挿入装置の構成図

【図13】実施の形態5における偽装コード挿入部および透かし挿入部の動作のフローチャート

【図14】実施の形態5における透かし挿入部が生成したプログラムの例

【図15】実施の形態5における透かし検出部の動作のフローチャート

【図16】本発明の実施の形態6の透かし挿入部の動作フローチャート

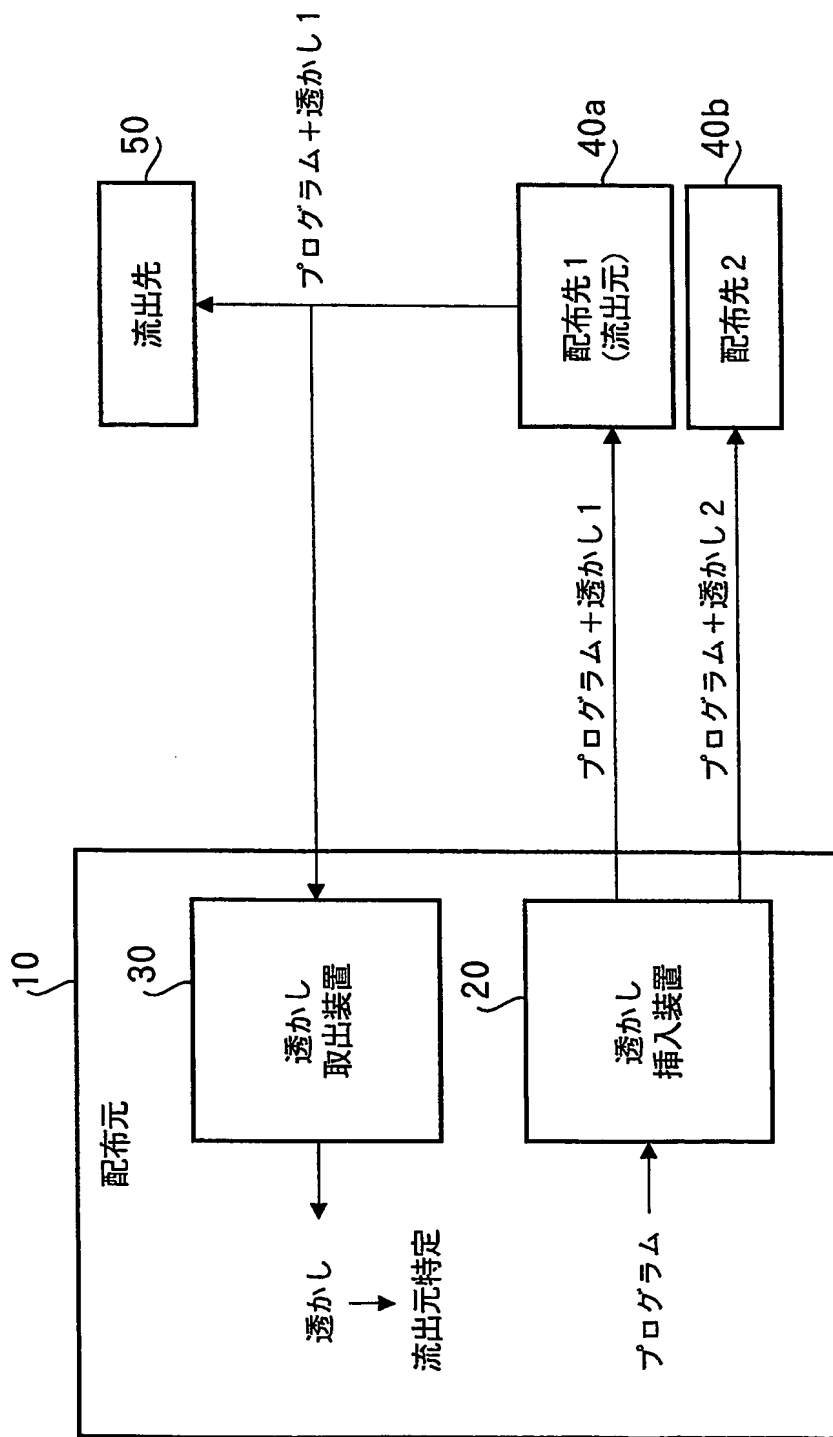
【図17】実施の形態6にかかる透かし挿入部が生成したプログラムを示す図

【符号の説明】

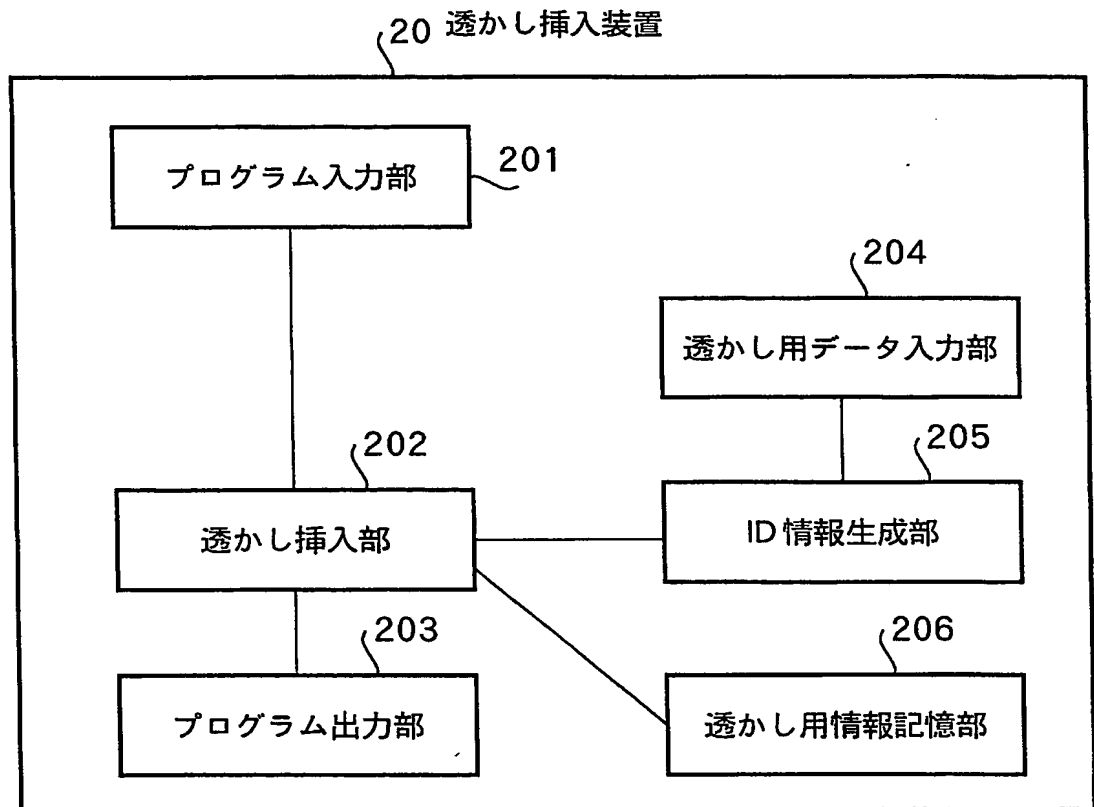
【0206】

- 10、1100 配布元
- 20、1200 透かし挿入装置
- 30 透かし取出装置
- 40a、40b、1110、1120 配布先
- 50、1130 流出先
- 201、301 プログラム入力部
- 202 透かし挿入部
- 203 プログラム出力部
- 204 透かし用データ入力部
- 205 ID情報生成部
- 206、305、1204 透かし用情報記憶部
- 302 透かし検出部
- 303 出力部
- 304 ID情報記憶部
- 1201 ダミーメソッド挿入部
- 1202 偽装コード挿入部
- 1203 ダミーメソッド入力部

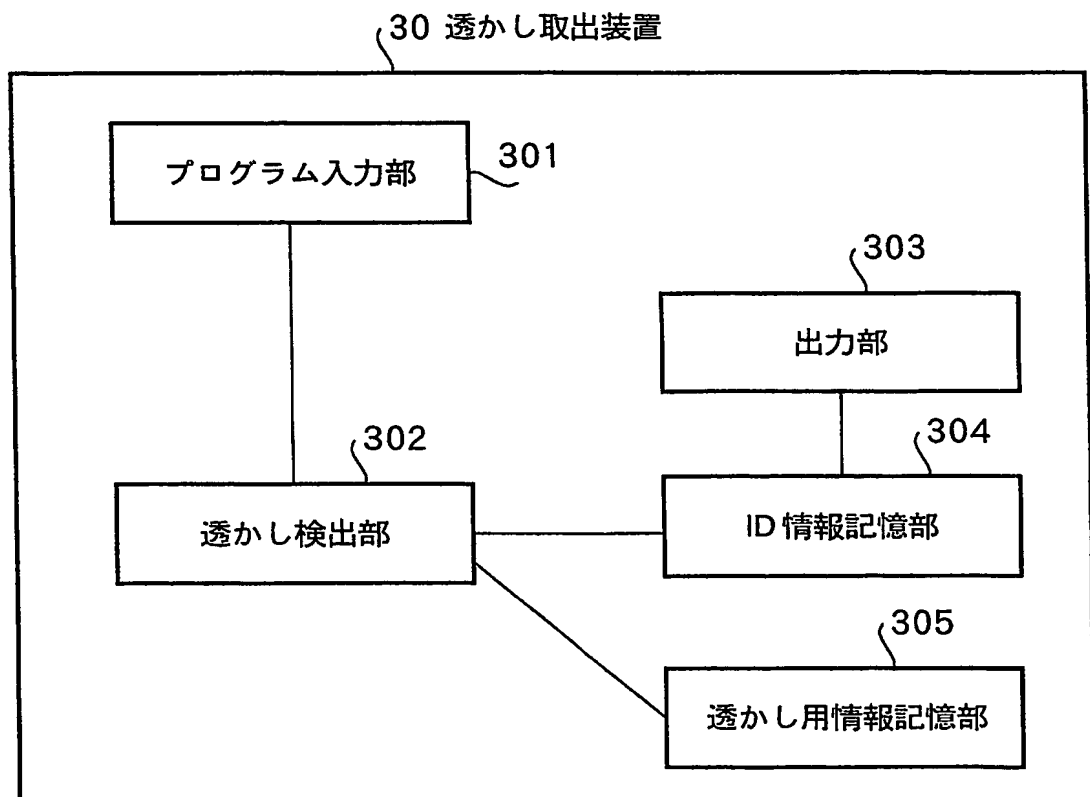
【書類名】図面
【図 1】



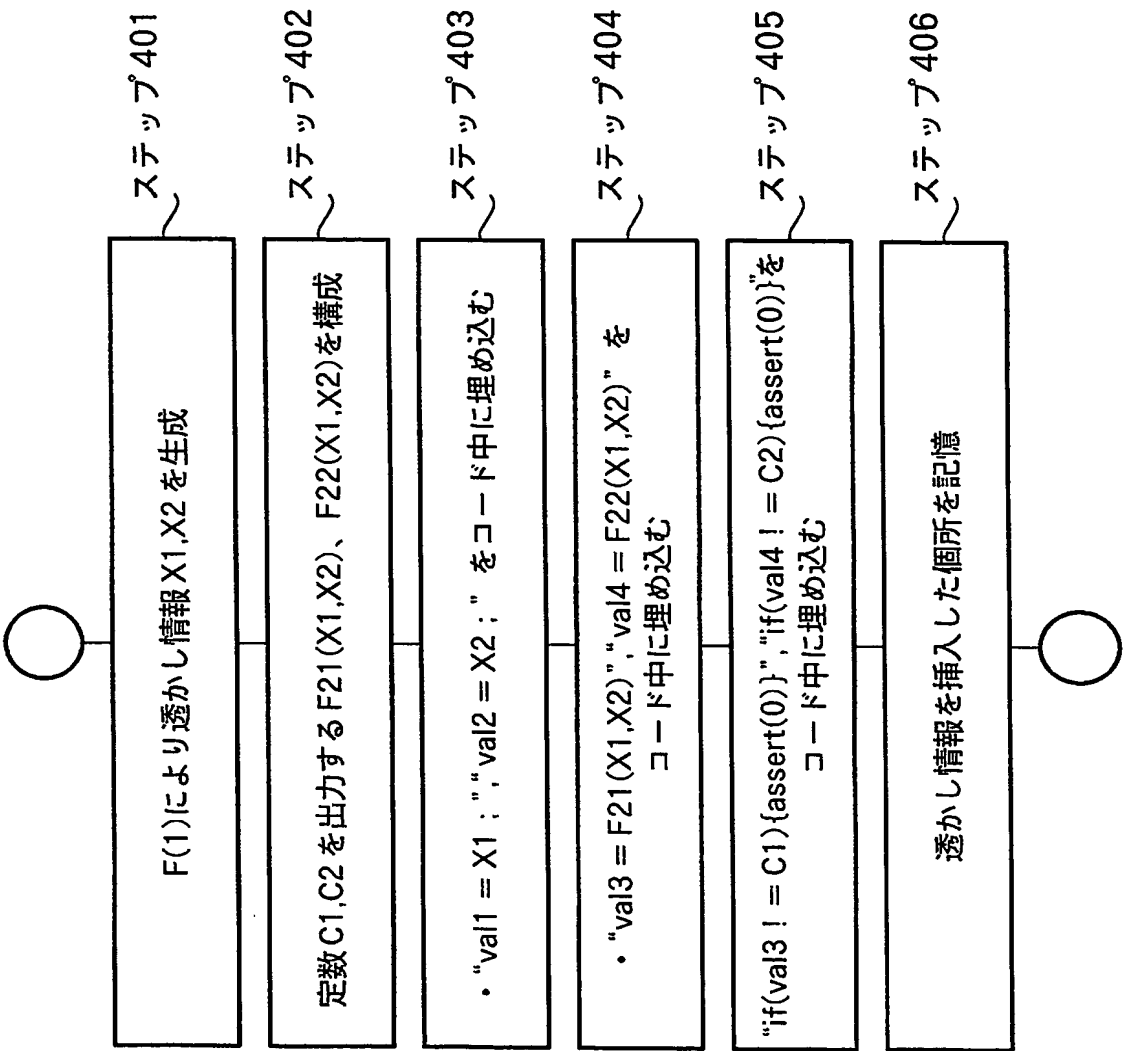
【図 2】



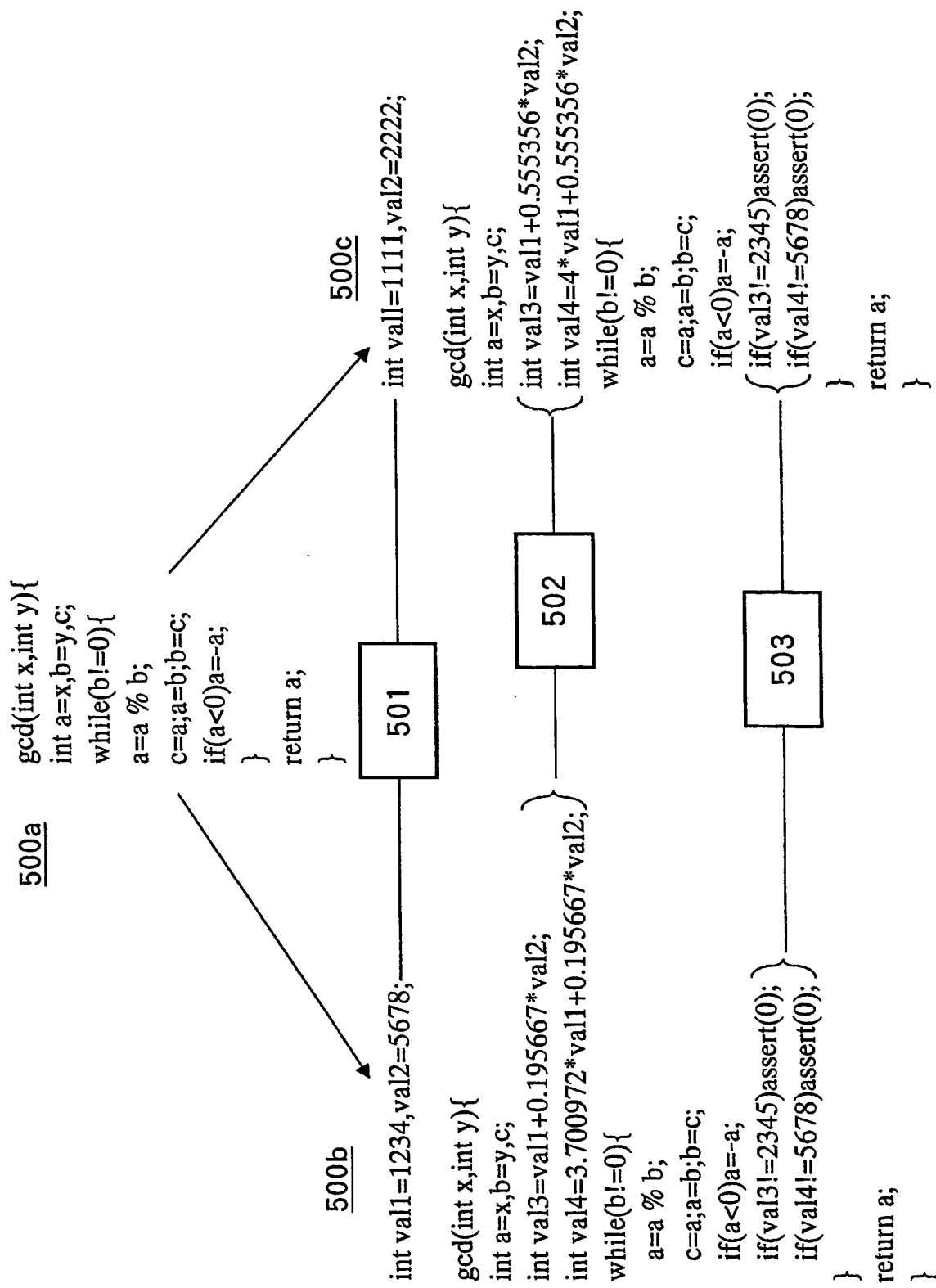
【図 3】



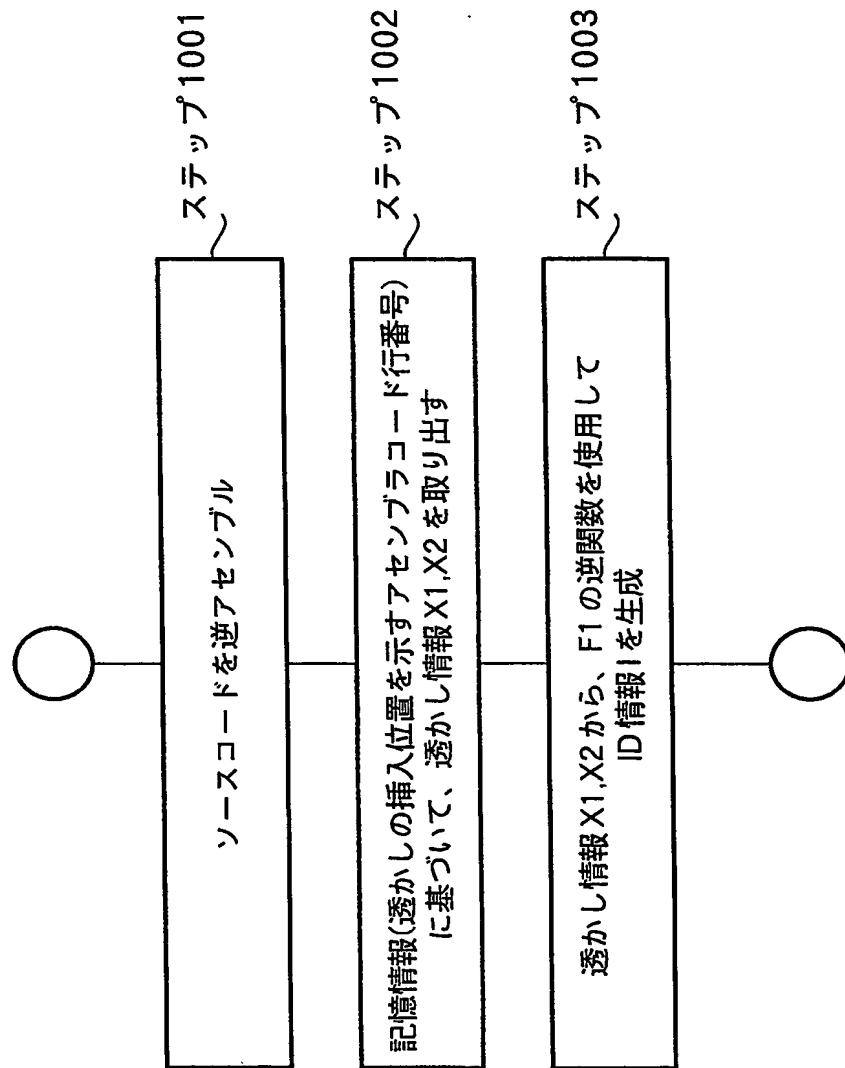
【図 4】



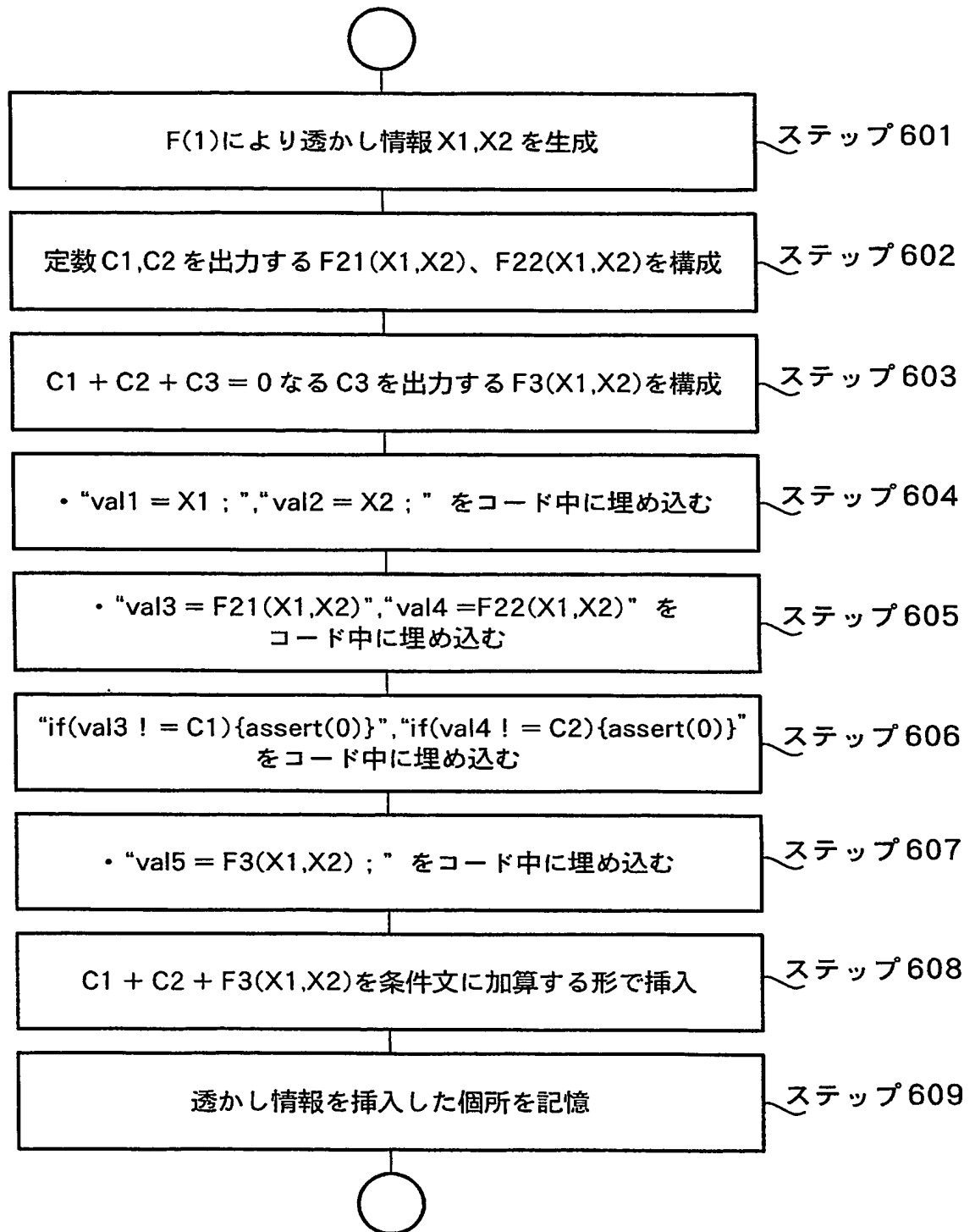
【図 5】



【図 6】



【図 7】



【図 8】

800a

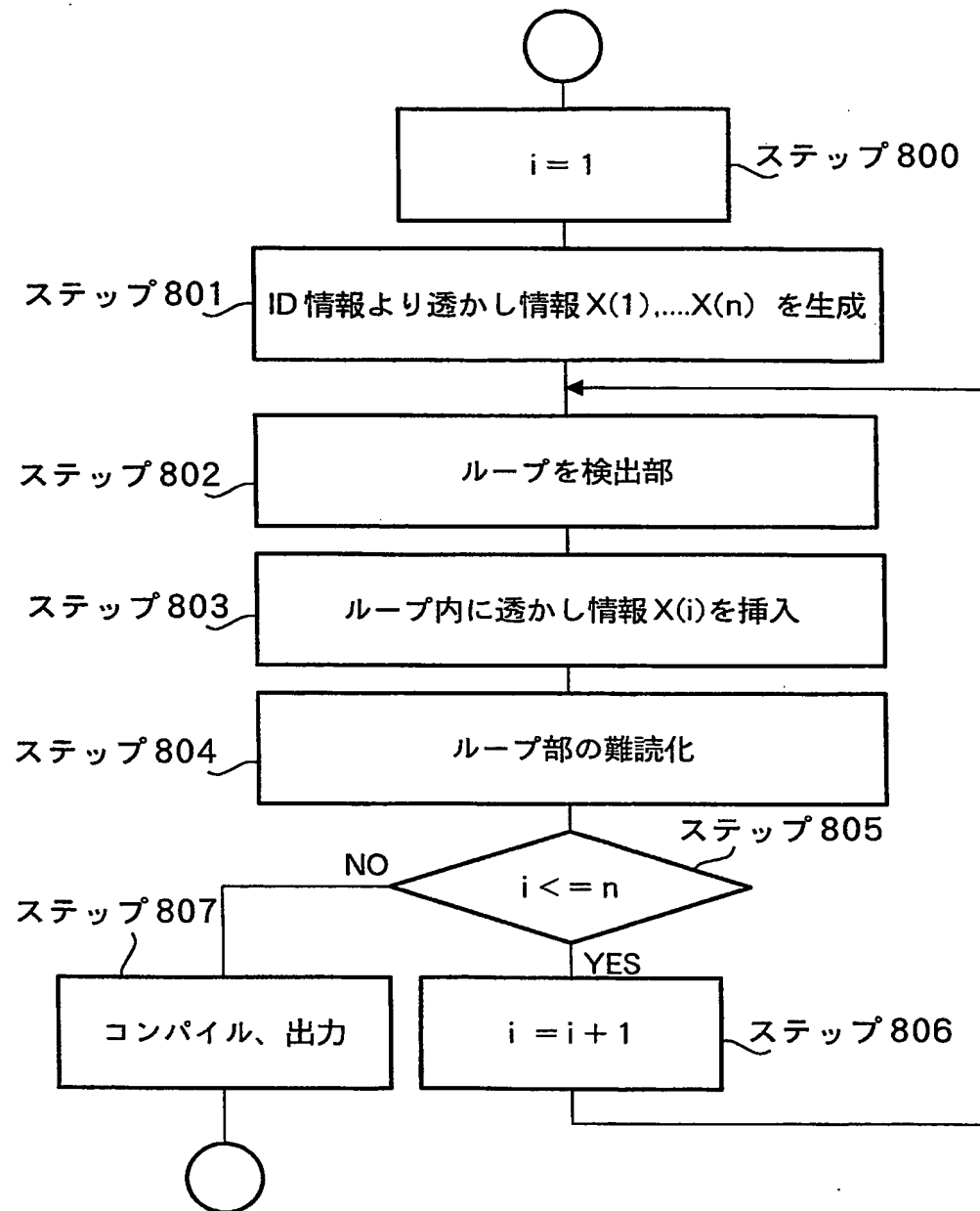
```
gcd(int x,int y){  
  int a=x,b=y,c;  
  while(b!=0){  
    a=a % b;  
    c=a; a=b; b=c;  
    if( a<0)a=-a;  
  }  
  return a;  
}
```

800b

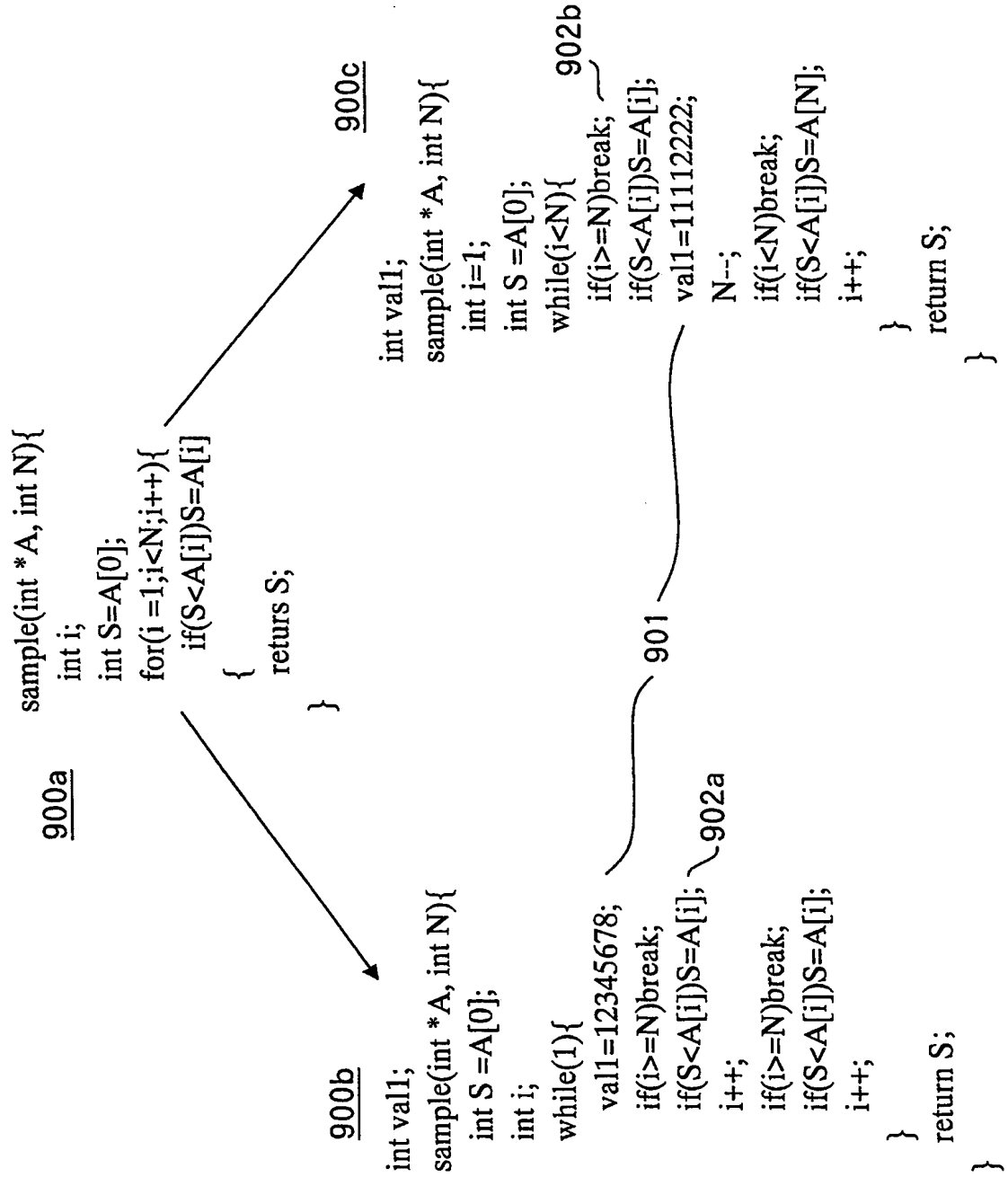
```
int val1=1234,val2=5678; ~701
```

```
gcd(int x,int y){  
  int a=x, b=y, c;  
  int val3=val1+0.195667*val2;  
  int val4=3.700972*val1+0.195667*val2;  
  int val5=5.601297*val1+0.195667*val2; } 702  
  while(b!=0){  
    a=a % b;  
    c=a;a=b;b=c;  
    if(a<val3+val4+val5)a=-a; ~ 703  
    if(val3!=2345)a++;  
    if(val4!=5678)assert(0); } 704  
  }  
  return a;  
}
```

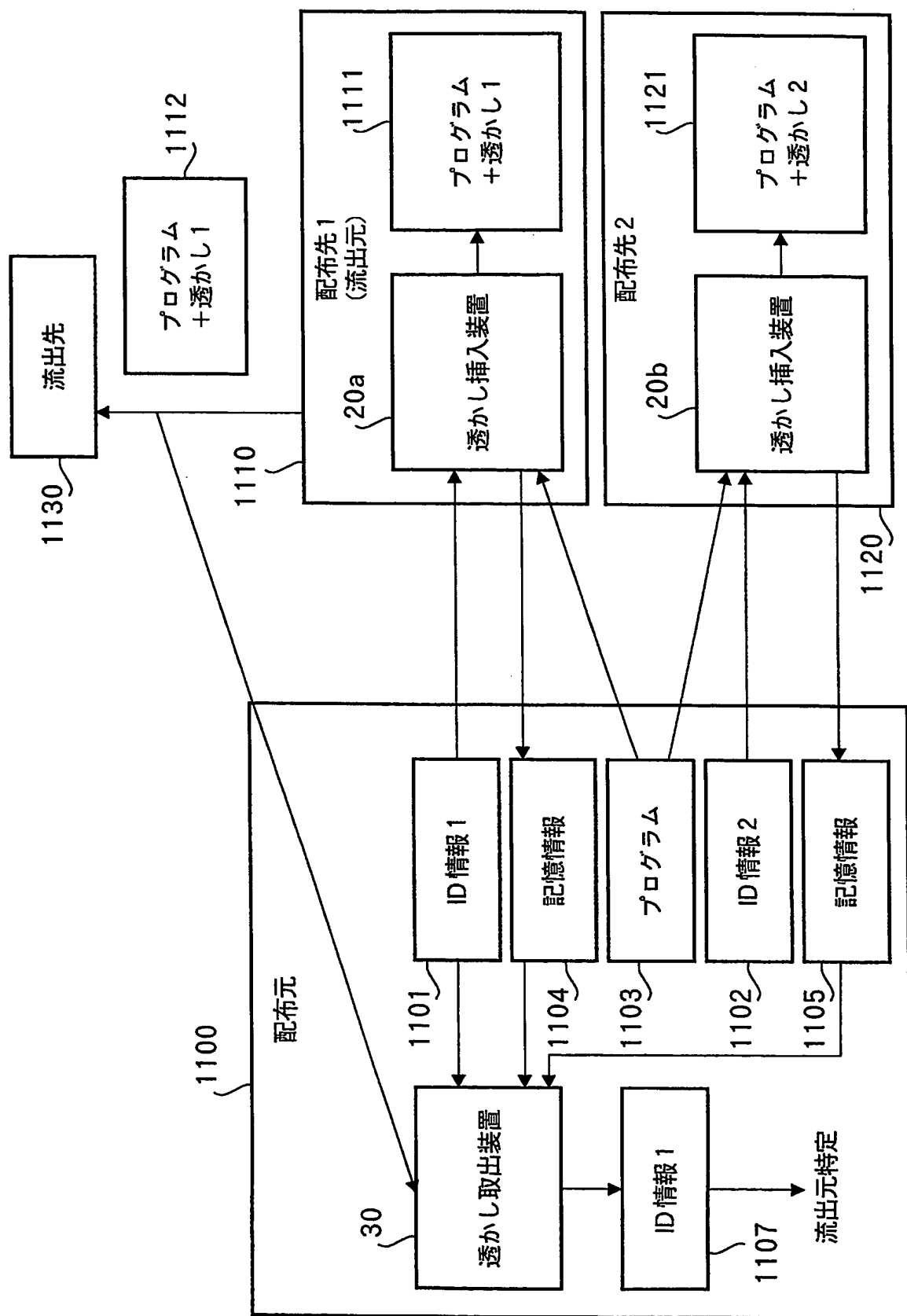
【図 9】



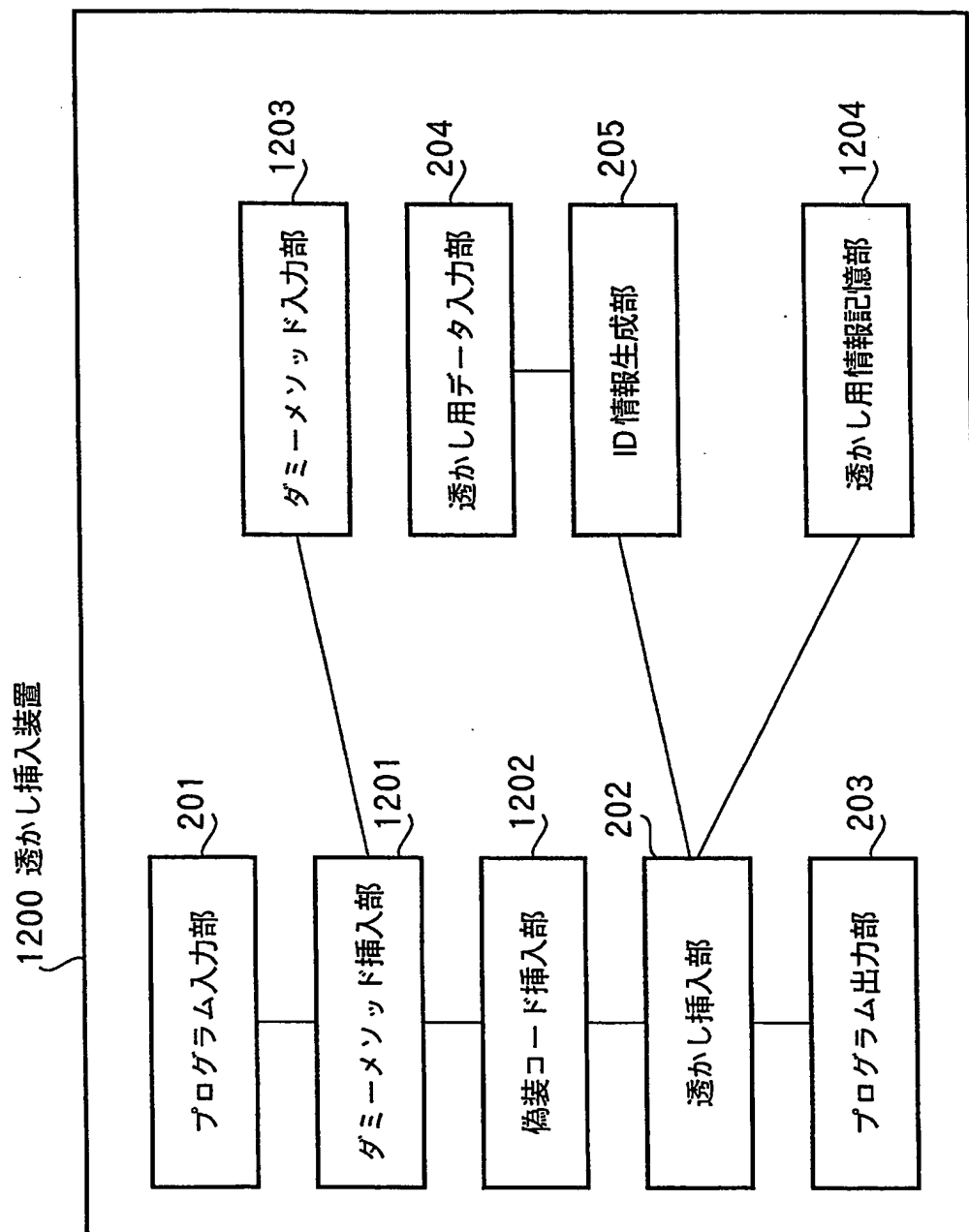
【図 10】



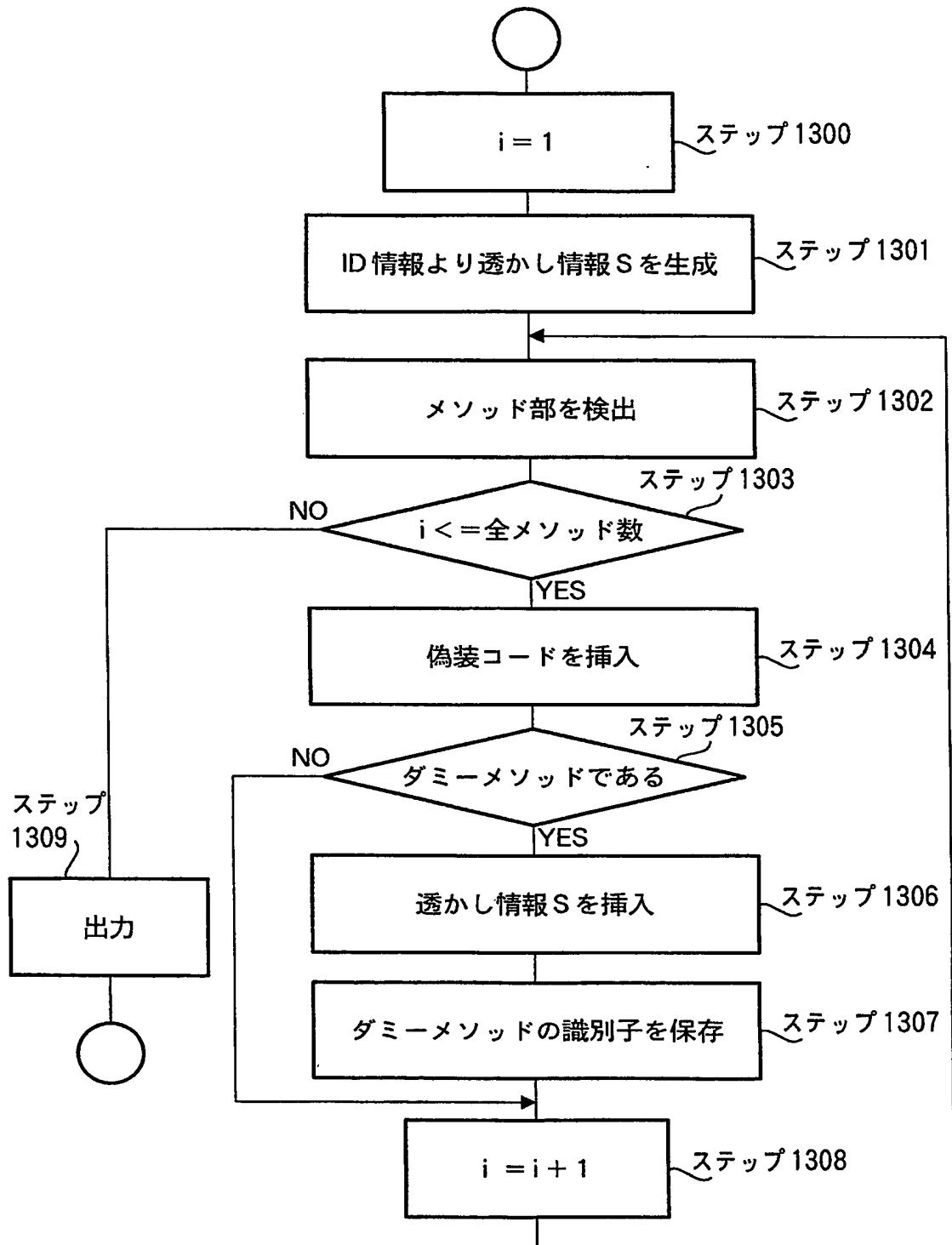
【図 11】



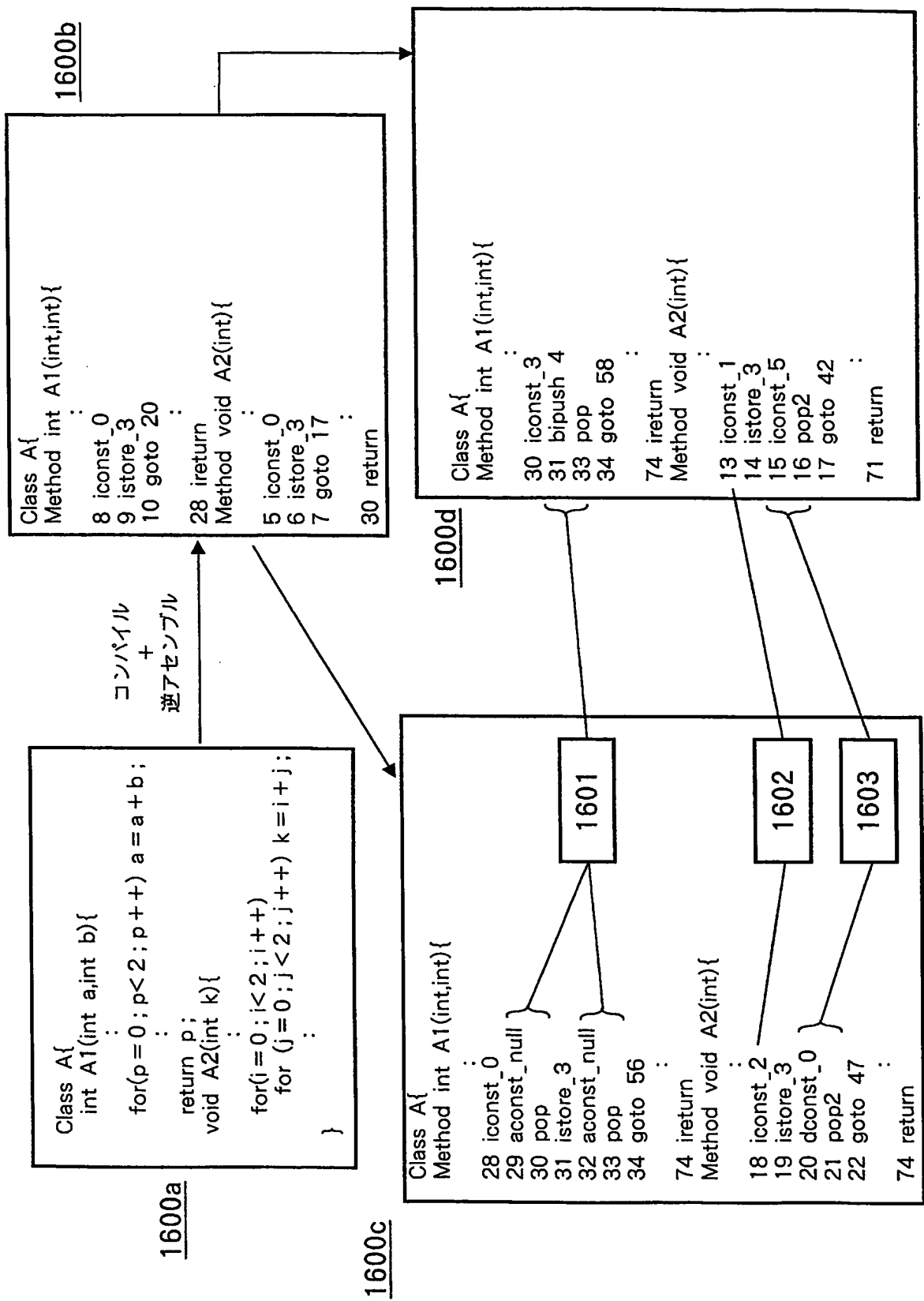
【図 12】



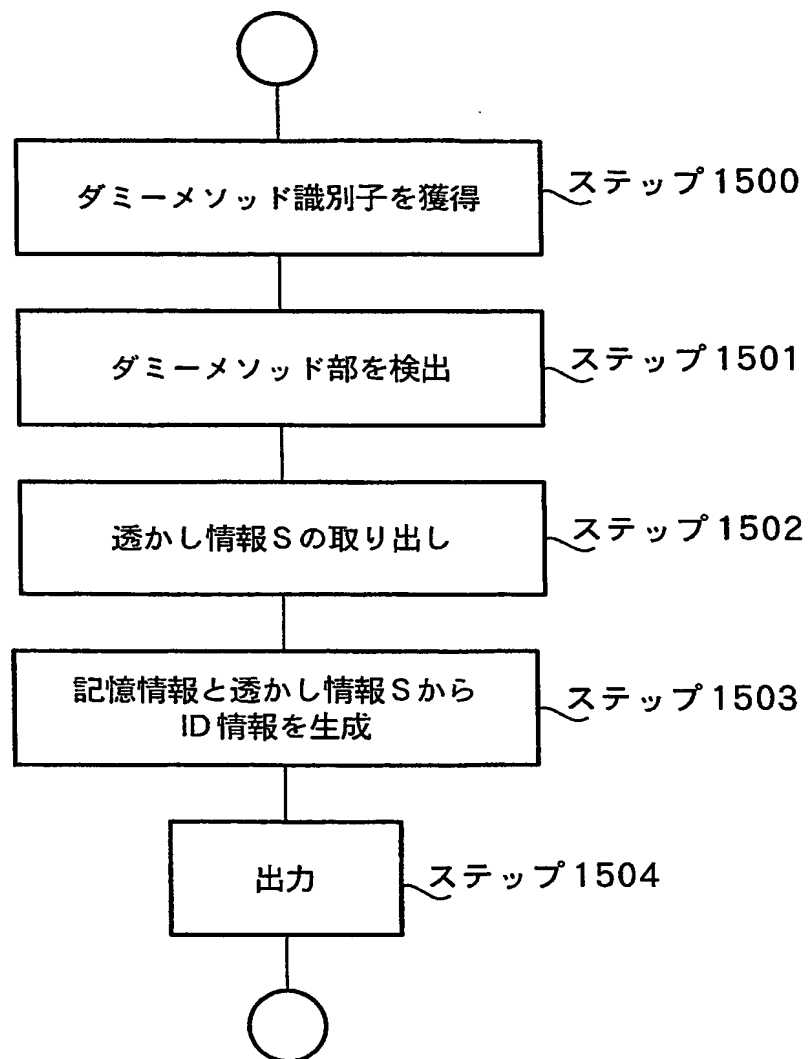
【図 13】



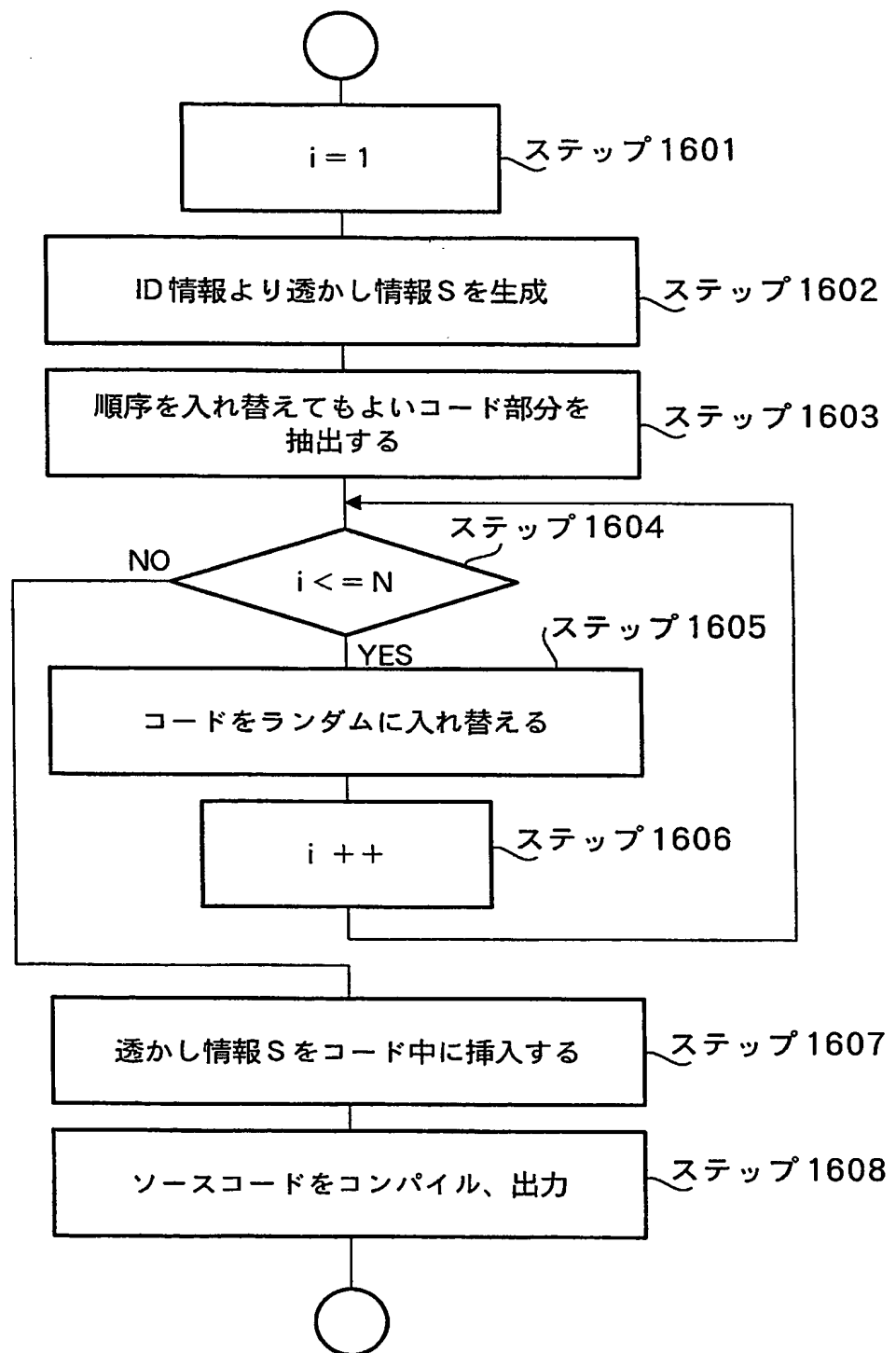
【図 14】



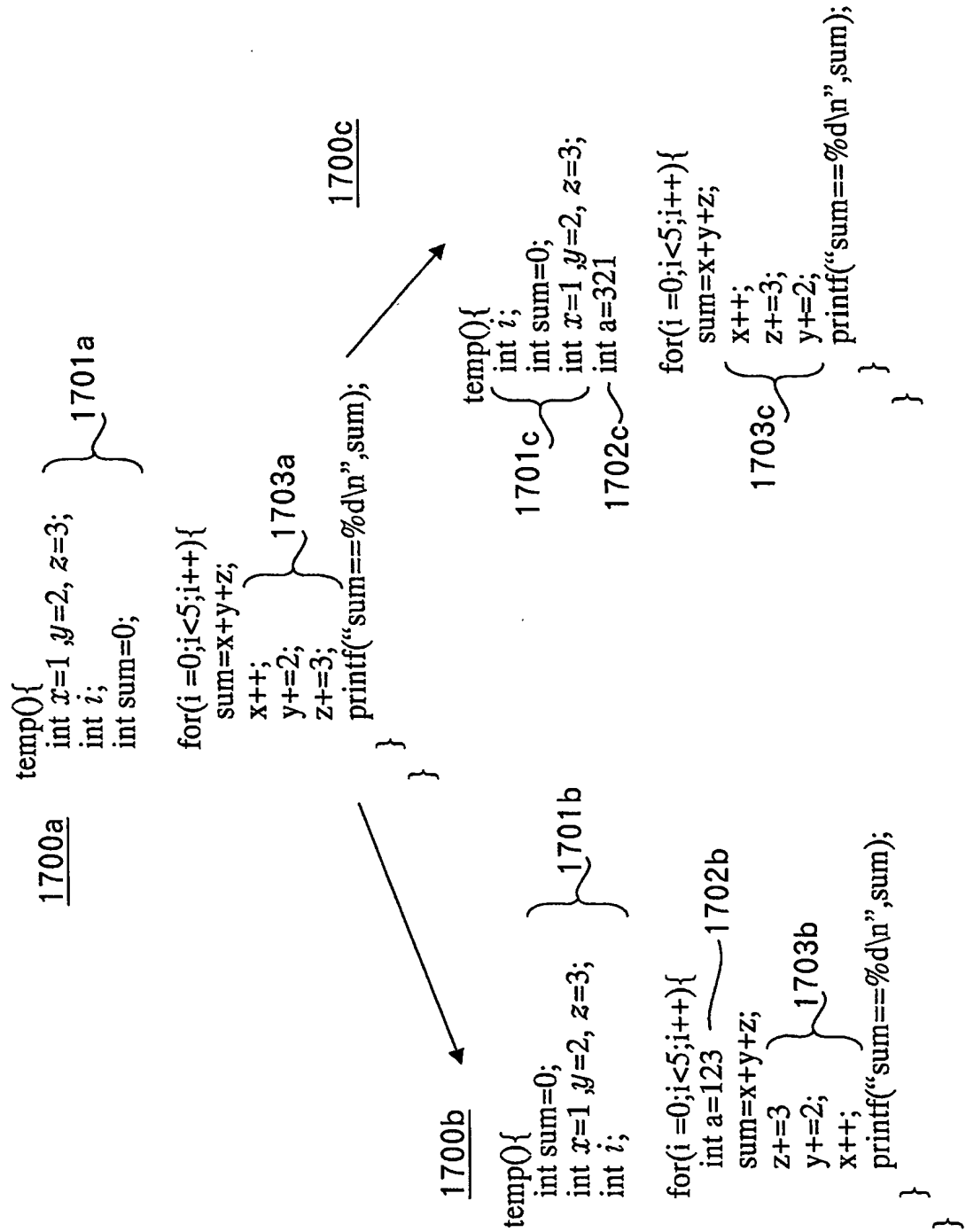
【図 15】



【図 16】



【図 17】



【書類名】要約書

【要約】

【課題】 透かしの挿入箇所を特定されないように透かしを挿入することにより、透かし情報がなく、かつ正常に動作するプログラムを生成できないようにすること。

【解決手段】 本発明は、プログラムの配布先を一意に特定するID情報から透かし情報を生成し、前記透かし情報を前記プログラムに挿入し、前記透かし情報挿入個所周辺や前記プログラム全体を前記プログラムの仕様が変更しないように前記配布先ごとに改変し、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させないものであり、前記配布先毎に異なるコードを前記プログラムに挿入するようにすることにより、差分攻撃により透かしであるコードが検出されないようにした。

【選択図】 図2

特願 2 0 0 3 - 3 2 4 8 0 5

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 8 2 1]

1. 変更年月日

1 9 9 0 年 8 月 2 8 日

[変更理由]

新規登録

住 所

大阪府門真市大字門真 1 0 0 6 番地

氏 名

松下電器産業株式会社